

UNCLASSIFIED

AD NUMBER

AD871086

LIMITATION CHANGES

TO:

Approved for public release; distribution is unlimited.

FROM:

Distribution: Further dissemination only as directed by Army Materiel Command, Washington, DC 20315, MAR 1970, or higher DoD authority.

AUTHORITY

AMC ltr 31 Aug 1971

THIS PAGE IS UNCLASSIFIED

AMSAA.TM 69



AMSAA

AD

TECHNICAL MEMORANDUM NO. 69

A LINEAR AND NONLINEAR SYSTEMS ANALYSIS TOOL:
LINE PRINTER PLOTS OF CHARACTERISTIC
EQUATION ROOT LOCI, BODE AND POPOV PLOTS OF
SYSTEM TRANSFER FUNCTIONS

by

Harold H. Burke
Robert L. Payne, Jr.

March 1970

This document may be further distributed by the holder only with specific prior approval of the Commanding General, U.S. Army Materiel Command, ATTN: AMCPA-SA, Washington, D.C. 20315.

U.S. ARMY ABERDEEN RESEARCH AND DEVELOPMENT CENTER
ARMY MATERIEL SYSTEMS ANALYSIS AGENCY
ABERDEEN PROVING GROUND, MARYLAND

AD871086L

AD871086



DISPOSITION

Destroy this report when no longer needed. Do not return it to the originator.

DISCLAIMER

The findings in this report are not to be construed as an official Department of the Army position.

WARNING

Information and data contained in this document are based on the input available at the time of preparation. The results may be subject to change and should not be construed as representing the AMC position unless so specified.

A R M Y M A T E R I E L S Y S T E M S A N A L Y S I S A G E N C Y

TECHNICAL MEMORANDUM NO. 69

MARCH 1970

A LINEAR AND NONLINEAR SYSTEMS ANALYSIS TOOL:
LINE PRINTER PLOTS OF CHARACTERISTIC
EQUATION ROOT LOCI, BODE AND POPOV PLOTS OF SYSTEM TRANSFER FUNCTIONS

Harold H. Burke

Robert L. Payne, Jr.

Combat Support Division

This document may be further distributed by the holder only with specific prior approval of the Commanding General, U.S. Army Materiel Command, ATTN: AMCPA-SA, Washington, D.C. 20315.

Project No. RDT&E 1P765801M1102

A B E R D E E N P R O V I N G G R O U N D , M A R Y L A N D

TECHNICAL MEMORANDUM NO. 69

HHBurke/RLPayne, Jr./flz
Aberdeen Proving Ground, Md.
March 1970

A LINEAR AND NONLINEAR SYSTEMS ANALYSIS TOOL:
LINE PRINTER PLOTS OF CHARACTERISTIC
EQUATION ROOT LOCI, BODE AND POPOV PLOTS OF SYSTEM TRANSFER FUNCTIONS

ABSTRACT

A computer program to perform dynamic systems analyses and plot the results is presented. Linear systems and a large classification of nonlinear systems representing engineering, scientific, and economic disciplines can be modeled to permit application of the computer program. Two examples are given to demonstrate the capabilities of the analysis tool. The mathematical model of a missile guidance and control system is analyzed and a ratio of polynomials representing the closed loop transfer function of a high performance model follower aircraft is evaluated. Linear differential equations to the 100th order having real or complex roots can be studied. System characteristic equation root loci and system transfer functions are plotted.

TABLE OF CONTENTS

ABSTRACT	3
LIST OF FIGURES	6
LIST OF TABLES	7
1. INTRODUCTION	9
2. THE PROBLEM	11
2.1 Linear System	11
2.2 Nonlinear System	15
3. GRAPHICAL METHODS	18
4. DATA FORMATS	19
4.1 General Description	19
4.2 Input Description	20
4.3 Output Description	
5. EXAMPLES	22
6. CONCLUSIONS	25
REFERENCES	55
APPENDIX A - LISTING OF SOURCE DECK	57
APPENDIX B - FLOW CHART OF SOURCE DECK	87
DISTRIBUTION LIST	229

LIST OF FIGURES

1. Linear Closed Loop System
2. Nonlinear Closed Loop System
3. Stability Boundary Popov Modified Phase Amplitude Characteristic
4. System Nonlinearity
5. Example 1, Linear System Bode Plot, Phase Angle
6. Example 1, Linear System Bode Plot, Magnitude
7. Example 1, Nonlinear System, Bode Plot, Phase Angle
8. Example 1, Nonlinear System, Bode Plot, Magnitude
9. Example 1, Linear System, Expanded Bode Plot (1 to 300 rad/s)
Phase Angle
10. Example 1, Linear System, Expanded Bode Plot (1 to 300 rad/s)
Magnitude
11. Example 1, Nonlinear System Expanded Bode Plot (1 to 300 rad/s)
Phase Angle
12. Example 1, Nonlinear System Expanded Bode Plot (1 to 300 rad/s)
Magnitude
13. Example 1, Linear System Expanded Bode Plot (120 to 200 rad/s)
Phase Angle
14. Example 1, Linear System Expanded Bode Plot (120 to 200 rad/s)
Magnitude
15. Example 1, Nonlinear System Expanded Bode Plot, (120 to 200 rad/s)
Phase Angle
16. Example 1, Nonlinear System Expanded Bode Plot, (120 to 200 rad/s)
Magnitude
17. Example 2, Linear System, Bode Plot, Phase Angle
18. Example 2, Linear System Bode Plot, Magnitude
19. Example 2, Nonlinear System, Bode Plot, Phase Angle
20. Example 2, Nonlinear System, Bode Plot, Magnitude
21. Example 2, Linear System, Expanded Bode Plot (1 to 200 rad/s),
Phase Angle
22. Example 2, Linear System, Expanded Bode Plot (1 to 200 rad/s),
Magnitude
23. Example 2, Nonlinear System, Expanded Bode Plot (1 to 200 rad/s),
Phase Angle
24. Example 2, Nonlinear System, Expanded Bode Plot (1 to 200 rad/s),
Magnitude

LIST OF TABLES

1. Input for Example 1.
2. Mirror of Input, Example 1.
3. Equivalent System Open Loop.
4. Standard Frequency Response and Popov Response, Example 1.
5. Mirror of Input, Example 2.
6. Equivalent System Open Loop Polynomials and Roots, Example 2.

1. INTRODUCTION

Systems analyses require that mathematical models be developed and exercised to determine cause-and-effect relationships. One powerful method is to generate linearized equations for each physical phenomenon occurring in the process and combine these into input-output relationships through the application of Laplace transformations.

Studying the characteristic equations of the input-output relations determines parametric stability trends.⁽¹⁾ Steady state frequency response of the system provides another useful characterization of system stability. Application of the inverse Laplace transformation to the process transfer functions provides transient histories of output for specific inputs.⁽²⁾

This report discusses a program that mechanizes another part of the analytical procedure, the determination of the frequency response of linear systems and the modified phase-amplitude characteristic for nonlinear systems. Reference 1, published in November 1968, discusses a program that mechanized the determination of the roots of the characteristic equation. Reference 2, discusses a program that determines the transient response of the process when it is subjected to input forcing functions.

Application of this computer program, which is an extension of the material contained in Reference 1, along with the computer program presented in Reference 2, permits one to determine quickly and accurately the stability and performance characteristics of high-order linear dynamic processes and to qualitatively predict the stability and performance of a large class of nonlinear systems.

The techniques developed in this report have been applied to specific studies.^{(3) (4)} A more extensive utilization of the methods will be contained in a forthcoming report which expands Chapter 4.2 of Reference 5. Preliminary work has been commenced to explore the utility of these methods in providing solutions to optimal inventory problems.⁽⁶⁾⁽⁷⁾

A useful linear system analysis method is to determine the steady state frequency response of the system. Evaluation of the polynomials representing the system's transfer function is a time-consuming task. Conversion of the tabulated results to a graphical display is then required before conclusions relevant to system stability can be appreciated.

The analysis of nonlinear systems is a difficult task, but for certain classes of systems it is possible to obtain qualitative indications of stability. Similarly, the task of obtaining the tabular data and converting it to a graphical form before conclusions can be drawn is a laborious process.

These two methods (one linear and one nonlinear) have been coded in a FORTRAN IV program and integrated into the root locus methods discussed in Reference 1. Main features of the program are:

1. FORTRAN IV program. No machine-oriented or object language.
2. No FORTRAN complex type statements necessary.
3. No special graphical plotting equipment.
4. Order of polynomial may be up to 100.
5. Number of variations of given parameter may be up to 100.

Main features of the graphical displays are:

ROOT LOCUS Method⁽¹⁾

1. Log plot of the third and fourth quadrants of the complex frequency plane from 0 to 10,000 radians/second.
2. Linear plots of selected regions of the third and fourth quadrants of the complex frequency plane with arbitrary scales.

LINEAR FREQUENCY RESPONSE Method

1. Plot of Magnitude (in decibels) and phase (in degrees) of transfer function vs 6 decades of frequency, beginning at a selectable minimum frequency.
2. Linear frequency plots of magnitude (in decibels) and phase (in degrees) of transfer function for selectable ranges of frequency.

NONLINEAR FREQUENCY RESPONSE Method

1. Plot of "Modified Magnitude Characteristic" (in decibels) and "Modified Phase Characteristic" (in degrees) of transfer function vs 6 decades of frequency, beginning at same selectable minimum frequency as linear frequency response method.

2. Linear frequency plots of "Modified Magnitude Characteristics" (in decibels) and "Modified Phase Characteristic" (in degrees) of transfer function for same selectable ranges of frequency as linear frequency response methods.

2. THE PROBLEM

2.1 Linear System.

Regardless of the complexity of a linear closed-loop system, its transfer function can be reduced to the equivalent form shown in Figure 1. For multiple loop systems, the G's and H's are readily expressed as sums of products of polynomials which are identified with individual elements making up the complete system.

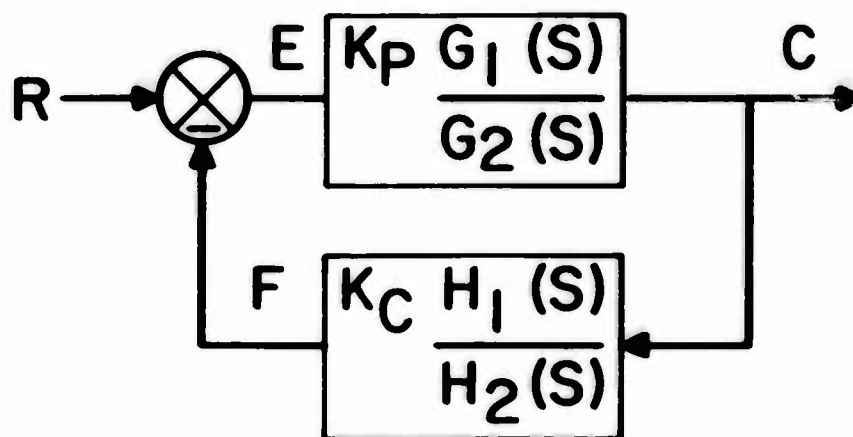


Figure 1. Linear Closed Loop System.

R = system input,
C = system output,
F = system feedback,
E = system error,
 K_p = process gain, and
 K_c = controller gain.

The fractions $G_1(s)/G_2(s)$ and $H_1(s)/H_2(s)$ are equivalent transfer fractions of the system and are represented by ratios of polynomials which upon expansion can be put into the form

$$\frac{\sum_{\mu=0}^{n-k} a_{\mu} s^{n-\mu}}{\sum_{i=0}^{m+n-l} b_i s^{m+n-i}}, \quad (1)$$

or in factored form

$$\frac{s^k \prod_{\mu=1}^v (s+z_{\mu}) \prod_{q=v+1}^y [s+(\sigma_q+j\omega_q)] [s+(\sigma_q-j\omega_q)]}{s^l \prod_{i=1}^g (s+p_i) \prod_{t=g+1}^r [s+(\sigma_t+j\omega_t)] [s+(\sigma_t-j\omega_t)]} \quad (2)$$

where $y = \frac{n-k-v}{2}$, $r = \frac{m+n-l-g}{2}$

where the a's and b's are real, the poles and zeros can be either real or complex in conjugate pairs, and $m \geq 1$.

There are two transfer functions of fundamental importance.

OPEN LOOP TRANSFER FUNCTION:

$$\frac{F}{E} = \left[K_p \frac{G_1(s)}{G_2(s)} \right] \left[K_c \frac{H_1(s)}{H_2(s)} \right]. \quad (3)$$

CLOSED LOOP TRANSFER FUNCTION:

$$\frac{C}{R} = \frac{K_p G_1(s) H_2(s)}{K_c K_p G_1(s) H_1(s) + G_2(s) H_2(s)}. \quad (4)$$

The linear system's closed-loop characteristic equation, which determines stability is

$$K_c K_p G_1(s)H_1(s) + G_2(s)H_2(s) = 0. \quad (5)$$

Dividing both sides of Equation (5) by $G_2(s)H_2(s)$ gives

$$K_c K_p \frac{G_1(s)H_1(s)}{G_2(s)H_2(s)} + 1 = 0 \quad (6)$$

If we let

$$K_c K_p = K^*, \quad (7)$$

$$G_1(s)H_1(s) = A(s), \text{ and} \quad (8)$$

$$G_2(s)H_2(s) = B(s), \text{ then Equation (6) becomes} \quad (9)$$

$$\frac{K^* A(s)}{B(s)} + 1 = 0. \quad (10)$$

In terms of the system open-loop transfer function,

$$\frac{F}{E} + 1 = 0. \quad (11)$$

This expression relates the system's closed-loop characteristic equation to the system's open-loop transfer function.

The steady state frequency response of the open-loop transfer is obtained by evaluating Equation (3) along the imaginary axis for values of ω between zero and infinity rad/s.

The result is a vector quantity,

$$\left. \frac{F}{E} \right|_{0 \leq \omega \leq \infty} = K^* \left. \frac{A(s)}{B(s)} \right|_{0 \leq s \leq \infty} = M_{oL} \angle \phi_{oL}, \quad (12)$$

where

M_{oL} = scalar magnitude of open-loop transfer function between

$0 \leq \omega \leq \infty$, and

ϕ_{oL} = direction of magnitude of open-loop transfer function

between $0 \leq \omega \leq \infty$.

Equation (10) can be rewritten as

$$M_{oL} \angle \phi_{oL} + 1 \angle 0^\circ = 0, \quad (13)$$

or

$$M_{oL} \angle \phi_{oL} = 1 \angle 180^\circ. \quad (14)$$

Equation 14 is the form required to provide quantitative stability information for a closed-loop system.

There are two figures of merit which provide a measure of system stability. The only restriction is that all roots of the system's open-loop transfer function denominator, $B(s)$, be ≤ 0 .

GAIN MARGIN is the amount that M_{oL} must be increased or decreased to make it equal to 1, only when the orientation of magnitude of the open-loop transfer function, $\angle \frac{A(s)}{B(s)}$, is 180° .

PHASE MARGIN is the amount that ϕ_{oL} must be increased or decreased to make it equal to 180° , only when the magnitude of the open-loop transfer function, $\left| \frac{A(s)}{B(s)} \right|$, is 1.

These concepts can be related directly to Equation 14. Incipient instability is identified with either a zero gain or zero phase margin.

The scalar magnitude of the open-loop transfer function is generally expressed in decibels. The decibel equivalent, $M_{oL} \text{ (db)}$, of a number, M_{oL} , is $M_{oL} \text{ (db)} = 20 \log_{10} M_{oL}$.

The gain margin and phase margin of a linear closed-loop system as shown in Figure 1 and Equation (4) can be readily determined by analyzing the open-loop transfer functions of Equation (3) in accordance with Equation (14).

2.2 Nonlinear System.

V.M. Popov's plots of system transfer functions give some insight into stability of nonlinear systems.⁽⁸⁾⁽⁹⁾ Figure 2 shows the system configurations considered in this analysis. It contains a linear part and one nonlinear element and is subject to the following restrictions.

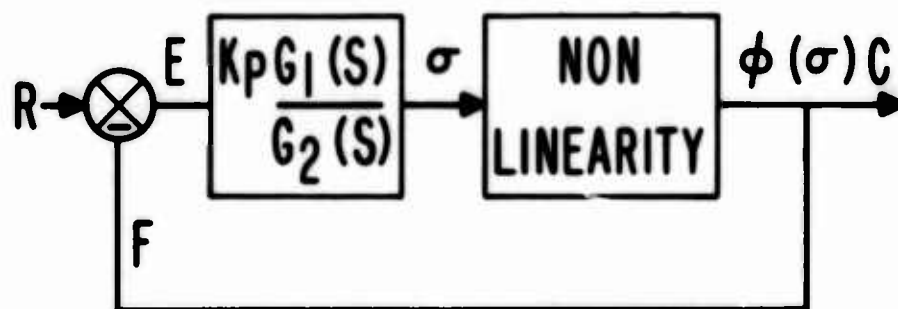


Figure 2. Nonlinear Closed Loop System

1. The roots of $G_2(s)$ have negative real parts and simple or multiple roots on the imaginary axis (one root permitted at origin).

$$2. \quad \epsilon \leq \frac{\phi(\sigma)}{\sigma} \leq K. \quad (15)$$

3. The linear system of Figure 2, obtained by substituting $\phi = \epsilon \sigma$, is stable.

If there exists a non-negative real number q such that

$$\operatorname{Re} \left[(1 + jq\omega) G(j\omega) \right] + \frac{1}{K} > 0 \quad (16)$$

for all $\omega > 0$, where G is defined by $\frac{G_1(s)}{G_2(s)}$, the nonlinear system is

asymptotically stable in the large.

Since G has been defined as the ratio of two polynomials, it may be expressed as

$$G(j\omega) = \frac{G_{1R}(\omega) + jG_{1I}(\omega)}{G_{2R}(\omega) + jG_{2I}(\omega)}, \quad (17)$$

where

G_{1R}, G_{2R} = real parts of G_1 , and G_2 , respectively, and

G_{1I}, G_{2I} = imaginary parts of G_1 and G_2 , respectively.

Redefining $G(j\omega)$, we have

$$G(j\omega) = G_R(\omega) + jG_I(\omega), \quad (18)$$

where

$$G_R = \frac{G_{1R}G_{2R} + G_{1I}G_{2I}}{(G_{2R})^2 + (G_{2I})^2}, \text{ and}$$

$$G_I = \frac{G_{1I}G_{2R} - G_{1R}G_{2I}}{(G_{2R})^2 + (G_{2I})^2}.$$

As a result, the inequality,

$$\operatorname{Re} \left[(1 + jq\omega) G(j\omega) \right] + \frac{1}{K} > 0, \quad (19)$$

becomes

$$G_R(\omega) - q\omega G_I(\omega) + \frac{1}{K} > 0. \quad (20)$$

To aid in a graphical interpretation of these results, we define

$$\begin{aligned} U &= G_R(\omega), \text{ and} \\ W &= \omega G_I(\omega). \end{aligned} \tag{21}$$

The inequality of Equation (20) may be written as

$$U - q W + \frac{1}{K} > 0, \tag{22}$$

where

$$q \geq 0, K > 0.$$

If we replace the inequality of Equation (22) by an equal sign,

$$U - q W + \frac{1}{K} = 0. \tag{23}$$

Then it may be seen that inequality of Equation (22) represents those points in the U-W plane of the modified phase amplitude characteristic which are to the right of the line of Equation 23, as shown in Figure 3.

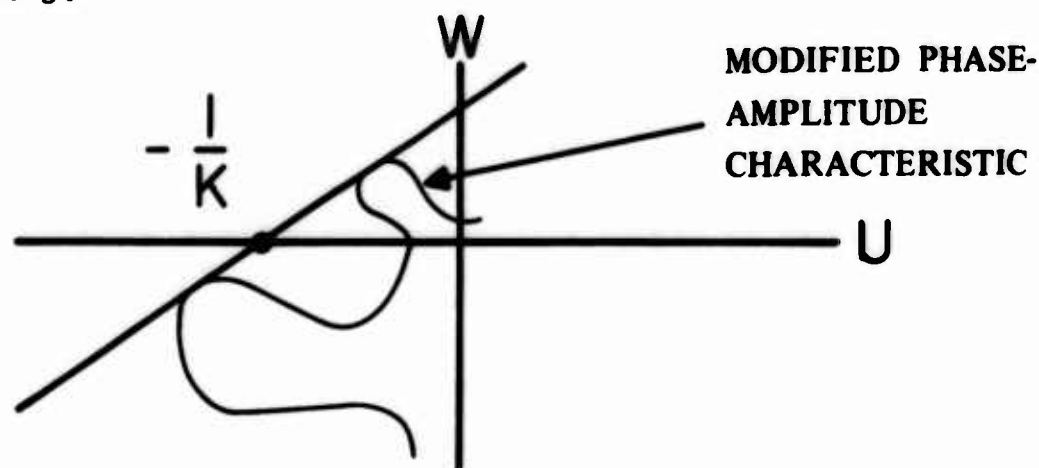


Figure 3. Stability Boundary-Popov Modified Phase Amplitude Characteristic.

The nonlinear system synthesis problem will be concerned with determining the variation of K for a given linear $G(s)$. Figure 4 shows the boundary constraints of the nonlinearity $\phi(\sigma)/\sigma$ of the system shown in Figure 2.

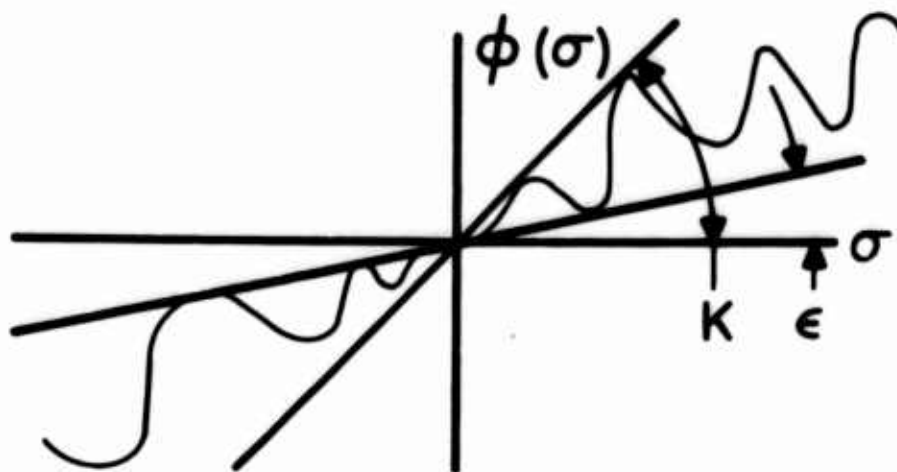


Figure 4. System Nonlinearity.

To maintain symmetry with the Bode Plots, the modified magnitude and phase characteristics are plotted on identical axes. Values of the modified amplitude associated with the modified phase characteristic for $90^\circ \leq \phi \leq 270^\circ$ (-90°) establish the contour described in Figure 3.

Once this contour is established in the U-W plane, the maximum K obtainable is determined by inspection from the intersection of the U axis and a straight line tangent to the modified phase amplitude characteristic.

3. GRAPHICAL METHODS

An on-line printout of the Bode diagram for an open-loop transfer function, F/E , or for a closed-loop transfer function, C/R , is generated for the linear system. Similarly, printouts of the Popov "modified amplitude-phase characteristics" are generated. The phase and amplitude axes are scaled in degrees and decibels with an arbitrarily selectable scale between $\pm \phi$ max degrees $\pm M$ max db, respectively. The frequency axes are logarithmically scaled in radians per second over a range of 6 decades, beginning with an arbitrarily selected lowest

decade. Values of amplitude and phase are calculated for 55 uniformly distributed frequencies over the 6-decade range, plus 17 additional points centered about each root of $\frac{A(s)}{K^*A(s) + B(s)}$. This additional distribution of printouts expressed as a multiple of each frequency is:

(0.01, 0.05, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8,
0.9, 1.0, 2.0, 4.0, 6.0, 8.0, 10).

An option of the program permits the density of the Bode plot to be either sparse or dense. The sparse plot requires $K^* = 0$ and only one set of frequencies are available, from which the amplitude and phase of $A(s)/B(s)$ are calculated. The dense plot permits K^* to have values consistent with the root locus subroutine.

For each value of K^* , a different set of frequencies will exist for $K^* A(s) + B(s)$.

A larger number of amplitude and phase values will be generated. Use of the dense plot capability may be necessary to obtain precise definition of the graphical plot in some cases.

Any number of simultaneous expands of different portions of the 6-decade frequency range may be plotted.

4. DATA FORMATS

4.1 General Description.

This is a subroutine designed to supplement the methods presented in AMSAA TM No. 21, A Linear Closed Loop System Analysis Procedure Using Line Printer Plots of Characteristic Equation Root Loci. This subroutine utilizes the polynomial multiplication and root locus methods discussed in AMSAA TM No. 21, and extends the analysis capabilities to include the following available options:

1. Root Locus Plots.
2. Frequency Response Plots.
3. Popov Modified Phase-Amplitude Characteristics Plots.

For the root locus option, the output is identical with the output obtained from AMSAA TM No. 21.

For the frequency-response option, the output is a log-log plot of steady state amplitude vs frequency and a semi-log plot of steady state phase angle vs frequency. Linear expands of amplitude and phase for any frequency range may be plotted.

For the Popov modified phase-amplitude characteristic, the output is a log-log plot of steady state amplitude vs frequency and semi-log plot of steady state phase angle vs frequency. Linear expands of amplitude and phase for any frequency range may be plotted.

4.2 Input Description.

a. Program option control card that precedes all data (see below).

b. Program option control card.

To run the program in its entirety with a dense plot in the frequency response, i.e.: Root Locus plus Expands and Frequency Response and Popov Plots plus expands place a one (1) in columns 10, 20, 30, and 40.

To run only the root locus log plot, place a one (1) in column 10 and zeros (0) in columns 20, 30, 40.

To run the root locus expand plot, place a one (1) in column 20 and zeros (0) in columns 10, 30, 40.

To run the sparse frequency response Popov plot only place a one (1) in column 30 and zeros in columns 10, 20, and 40.

To run the dense frequency response plot only, place a one (1) in columns 30 and 40 and zeros in columns 10 and 20.

To void all plot routines place a zero in columns 10, 20, 30, and 40.

c. Comment Card that follows the program option control card for identification purposes.

d. Program Data Cards.

Same format as described in Chapter IV (Data Formats), paragraph A2 of AMSAA TM No. 21. Control Card used to call All Plot Subroutines is same as control card used to call log plot subroutine of AMSAA TM No. 21 (Section IV, paragraph B2).

Control cards used to call root locus option are same as control cards used to call root locus expand option, paragraph B2, section IV of AMSAA TM No. 21. Table 1 demonstrates this input format for the dense frequency response option. Control cards used to call frequency response, Popov plots, and expands of both are additional features of this report.

<u>DESCRIPTION</u>	<u>COLUMNS</u>	<u>DATA</u>
Data appears only once	a. - 1-10	Smallest frequency, rad/s (real)
	b. - 11-20	Max phase angle, deg, (real)
	c. - 21-30	Max amplitude, DB (real)
Frequency response, Popov	a. - 1-10	<u>Zero</u> , if no expands desired (integer), One, if expands desired
Number of expands desired	a. - 1-10	No. of freq, response phase angle expands (integer)
	b. - 11-20	No. of freq response amplitude expands (integer)
	c. - 21-30	No. of Popov phase angle expands (integer)
	d. - 31-40	No. of Popov amplitude expands (integer)
Region of expand, one card for each expand in consecutive order with a,b,c,d, above.	1-10	Smallest freq, rad/s (real)
	11-20	Largest freq, rad/s (real)

4.3. Output Description.

Depending on the options specified, the output consists of:

Root Locus Plot Option

- a. A list of all of the roots plotted on the complex frequency plane log plot.
- b. Two log plots of the roots calculated by the polynomial multiplication and root locus program.
- c. The specified number of linear expand plots of selectable regions in the complex frequency plane.

FREQUENCY RESPONSE/POPOV PLOT OPTION

- a. A list of amplitude, amplitude in decibels, and phase angle in degrees for various frequencies specified in Section 3.
- b. A log-log plot of amplitude vs frequency and a semi-log plot of phase angle vs frequency.
- c. The specified number of linear expands of the frequency axis for the amplitude and phase angles of the specified transfer function.

5. EXAMPLES

The two examples discussed in AMSAA TM No. 21 will be analyzed, using the frequency response/Popov plot option. The root loci of $A(s)/B(s)$ using K^* as a parameter have previously been determined. The characteristic equation of the system shown in Figure 2 of Reference 1 is

$$K^*A(s) + B(s) = 0.$$

The open-loop transfer function for this expression, as given in Equation (12) is

$$\frac{F}{E} = \frac{K^*A_1(s) + A_2(s) + A_3(s)}{B(s)} \quad (24)$$

EXAMPLE 1.

where

$$K^* = -0.005$$

$$A_1(s) = 2092(s^2 + 1.2s - 1672)(s^2 + 3.2s + 25,600) + 491s^2(s^2 + 6s + 400)$$

$$A_2(s) = 0.081(464)(s + 2.7)(s^2 + 3.2s + 25,600)$$

$$A_3(s) = 0.95(464)(s + 2.7)(s^2 + 3.2s + 25,600)$$

$$B(s) = s(s^2 + 6s + 400)(s^2 + 3.2s + 25,600)$$

Table 1 shows the input card format for this open-loop transfer function.

Lines 2 through 24 apply to the polynomial multiplication and root locus methods and lines 26 through 36 apply to the frequency response and Popov condition plot subroutines.

The basic tabulated numerical output is shown in Tables 2 and 3 and is similar to Tables 2 and 3 of Reference 1. A tabulation of the open-loop frequency response gain and phase and Popov modified phase amplitude characteristics are shown in Table 4. Figures 5 and 6 are the Bode plots of F/E and Figures 7 and 8 are the Popov plots of F/E . Two regions are expanded to demonstrate capability of this program. First the regions of frequencies between 1.0 rad/s and 300 rad/s are shown in Figures 9 and 10 (frequency response), 11 and 12 (Popov response) and secondly the region of frequencies between 120 rad/s and 200 rad/s shown in Figures 13 and 14 (frequency response), 15 and 16 (Popov response) are plotted. Slope information consistent with Bode principles may be applied directly to these plots.

TABLE 1. INPUT FOR EXAMPLE 1

1-10,	11-20	COLUMNS		Remarks
		21-30	31-40	
0	0	1	1	Program option control card
Frequency Response				Comment card
0	4	1	504	Problem data cards
0	1×10^{-9}	1×10^{-9}		(Polynomial multiplication and Root Locus program) Same format as AMSAA TM No. 21, Chapter IV, Paragraph A2.
4	3	4	4	
3				
1	.005			
3	25600.	3.2	1.	
3	-1672.	1.2	1.	
1	-2092.			
1	.005			
3	0.	0.	-491.5	
3	400.	6.	1.	
1	.081			
2	0.	464.		
2	2.7	1.		
3	25600.	3.2	1.	
1	.95			
1	464.			
2	2.7	1.		
3	25600.	3.2	1.	
2	0.	1.		
3	400.	6.	1.	
3	25600.	3.2	1.	
0	0	0	10000	Same as paragraph B2, Chapter IV, TM No. 21. Control card used to call all plot sub-routines.
.01	180.	100.		Frequency response and Popov response plot subroutine.
1				
2	2	2	2	
1.	300.			
120.	200.			
1.	300.			
120.	200.			
1.	300.			
120.	200.			
1.	300.			
120.	200.			

EXAMPLE 2

Table 6, and Figures 12 through 14 of Reference 1 are the $A(s)/B(s)$ for a 24th order linear system. Table 5 gives the F/E data input for this system, and Table 6 lists open-loop transfer function. Figures 17 and 18 show the Bode plot and Figures 19 and 20 show the Popov plot. An expand plot of the region between $1 \text{ rad/s} \leq \omega \leq 200 \text{ rad/s}$ is shown in Figures 21 and 22 for the frequency response and in Figures 23 and 24 for the Popov response.

These two examples demonstrate the significant systems analysis capability that can be realized by using the tools developed in this document and the related programs contained in References 1 and 2. Detailed conclusions can be realized when high order feedback control systems are encountered.

6. CONCLUSIONS

The method described in this memorandum for computing linear system root loci, steady state amplitude and phase, and the nonlinear system steady state modified phase-amplitude characteristic has the following merits:

1. Accepts input in the form of sums of products of polynomials.
2. Produces a log plot of the entire complex plane in two graphical displays.
3. Produces selectable scaled linear plots of regions in complex frequency plane.
4. Relates the closed-loop gain to the graphical display.
5. Produces a magnitude (db) vs log frequency steady state response plot.
6. Produces a phase (deg) vs log frequency steady state response plot.
7. Produces selectable scaled linear plots of magnitude and phase of the steady state frequency response.

TABLE 2. MIRROR OF INPUT, EXAMPLE 1.

POLYNOMIAL MULTIPLICATION AND ROOT LOCUS

DELTA=		0	POLY. ADDED IN A(S)=	4	POLY. ADDED IN B(S)=	1	PROB. NO.=	504
K-INITIAL=								
NUMBER OF POLY. IN GROUP 1 OF NUMERATOR=								
NUMBER OF POLY. IN GROUP 2 OF NUMERATOR=								
NUMBER OF POLY. IN GROUP 3 OF NUMERATOR=								
NUMBER OF POLY. IN GROUP 4 OF NUMERATOR=								
NUMBER OF POLY. IN GROUP 1 OF DENOMINATOR=								
C1 1)=		.0050000000						
C1 1)=		25600.0000000000						
C1 2)=		3.2000000000						
C1 3)=		1.0000000000						
C1 1)=		-1672.0000000000						
C1 2)=		1.2000000000						
C1 3)=		1.0000000000						
C1 1)=		-2092.0000000000						
C1 1)=		.0050000000						
C1 1)=		.0000000000						
C1 2)=		.0000000000						
C1 3)=		-491.5000000000						
C1 1)=		400.0000000000						
C1 2)=		6.0000000000						
C1 3)=		1.0000000000						
C1 1)=		.0810000000						
C1 1)=		.0000000000						
C1 2)=		464.0000000000						
C1 1)=		2.7000000000						
C1 2)=		1.0000000000						
C1 1)=		25600.0000000000						
C1 2)=		3.2000000000						
C1 3)=		1.0000000000						
C1 1)=		.9500000000						
C1 1)=		464.0000000000						
C1 1)=		2.7000000000						
C1 2)=		1.0000000000						
C1 1)=		25600.0000000000						
C1 2)=		3.2000000000						
C1 3)=		1.0000000000						
C1 1)=		.0000000000						
C1 2)=		1.0000000000						
C1 1)=		400.0000000000						
C1 2)=		6.0000000000						
C1 3)=		1.0000000000						
C1 1)=		25600.0000000000						
C1 2)=		3.2000000000						
C1 3)=		1.0000000000						

TABLE 3. EQUIVALENT SYSTEM OPEN LOOP POLYNOMIALS
AND ROOTS, EXAMPLE 1.

COEFFICIENTS ARE GIVEN IN ASCENDING ORDER

THE COEFFICIENTS OF POLYNOMIAL A (ORDER = 4)			
4.7818957E 08	1.3620729E 07	7.1376580E 05	6.0177660E 02
			2.4666500E 01
THE ROOTS OF A			
-9.7472771E 00	+1	2.4364008E 01	
-2.4509797E 00	+1	-1.6776908E 02	
			-2.4509797E 00
			+1
			1.6776908E 02
THE COEFFICIENTS OF POLYNOMIAL B (ORDER = 5)			
0.0000000E 00	1.0240000E 07	1.5488000E 05	2.6019200E 04
			9.2000000E 00
THE ROOTS OF B			
0.0000000E 00	+1	0.0000000E 00	
-1.6000000E 00	+1	-1.5999200E 02	
			-3.0000000E 00
			+1
			1.9773720E 01
			+1
			1.5999200E 02
			-3.0000000E 00
			+1
			-1.9773720E 01
			1.0000000E 00

TABLE 4. STANDARD FREQUENCY RESPONSE AND POPV RESPONSE,
EXAMPLE 1.

OMEGA	/M/	/M/-DB	PMI	/M/P	/M/P-DB	PMI-P
1.00000E-02	4.66982E 03	7.33860E 01	-8.99923E 01	4.67023E 01	3.33868E 01	-8.92385E 01
2.00000E-02	2.33491E 03	6.73654E 01	-8.99847E 01	4.67024E 01	3.33868E 01	-8.92366E 01
3.00000E-02	1.55661E 03	6.38436E 01	-8.99770E 01	4.67024E 01	3.33868E 01	-8.92359E 01
4.00000E-02	1.16746E 03	6.13448E 01	-8.99694E 01	4.67024E 01	3.33868E 01	-8.92356E 01
5.00000E-02	9.33967E 02	5.94066E 01	-8.99617E 01	4.67025E 01	3.33868E 01	-8.92354E 01
6.00000E-02	7.78307E 02	5.78230E 01	-8.99541E 01	4.67026E 01	3.33868E 01	-8.92353E 01
7.00000E-02	6.67122E 02	5.64841E 01	-8.99464E 01	4.67026E 01	3.33868E 01	-8.92352E 01
8.00000E-02	5.83733E 02	5.53243E 01	-8.99388E 01	4.67027E 01	3.33868E 01	-8.92351E 01
9.00000E-02	5.18875E 02	5.43012E 01	-8.99311E 01	4.67028E 01	3.33869E 01	-8.92351E 01
1.00000E-01	4.66988E 02	5.33861E 01	-8.99235E 01	4.67029E 01	3.33869E 01	-8.92350E 01
2.00000E-01	2.33504E 02	4.73659E 01	-8.98469E 01	4.67047E 01	3.33872E 01	-8.92348E 01
3.00000E-01	1.55679E 02	4.38446E 01	-8.97704E 01	4.67076E 01	3.33878E 01	-8.92348E 01
4.00000E-01	1.16771E 02	4.13467E 01	-8.96938E 01	4.67117E 01	3.33885E 01	-8.92347E 01
5.00000E-01	9.34277E 01	3.94095E 01	-8.96173E 01	4.67170E 01	3.33895E 01	-8.92347E 01
6.00000E-01	7.78679E 01	3.78272E 01	-8.95408E 01	4.67234E 01	3.33907E 01	-8.92347E 01
7.00000E-01	6.67556E 01	3.64897E 01	-8.94643E 01	4.67310E 01	3.33921E 01	-8.92347E 01
8.00000E-01	5.84229E 01	3.53317E 01	-8.93878E 01	4.67398E 01	3.33937E 01	-8.92348E 01
9.00000E-01	5.19433E 01	3.43106E 01	-8.93112E 01	4.67498E 01	3.33956E 01	-8.92348E 01
1.00000E 00	4.67609E 01	3.33977E 01	-8.92348E 01	4.67609E 01	3.33977E 01	-8.92348E 01
2.00000E 00	2.34755E 01	2.74123E 01	-8.84708E 01	4.69385E 01	3.34306E 01	-8.92352E 01
3.00000E 00	1.57581E 01	2.39501E 01	-8.77089E 01	4.72408E 01	3.34863E 01	-8.92360E 01
4.00000E 00	1.19354E 01	2.15367E 01	-8.69521E 01	4.76782E 01	3.35664E 01	-8.92374E 01
5.00000E 00	9.67357E 00	1.97117E 01	-8.62039E 01	4.82660E 01	3.36728E 01	-8.92397E 01
6.00000E 00	8.19588E 00	1.82719E 01	-8.54702E 01	4.90260E 01	3.38085E 01	-8.92435E 01
7.00000E 00	7.17053E 00	1.71110E 01	-8.47602E 01	4.99883E 01	3.39774E 01	-8.92494E 01
8.00000E 00	6.43296E 00	1.61682E 01	-8.40883E 01	5.11943E 01	3.41844E 01	-8.92585E 01
9.00000E 00	5.89331E 00	1.54072E 01	-8.34762E 01	5.27006E 01	3.44363E 01	-8.92720E 01
1.00000E 01	5.49962E 00	1.48066E 01	-8.29572E 01	5.45854E 01	3.47415E 01	-8.92922E 01
2.00000E 01	5.49063E 00	1.47924E 01	-1.26454E 02	5.83866E 01	3.89277E 01	-9.21153E 01
3.00000E 01	1.06161E 00	5.19260E-01	-1.40243E 02	8.83866E 01	2.61860E 01	-9.22945E 01
4.00000E 01	6.8701E-01	-3.49537E 00	-1.19123E 02	2.33687E 01	2.73727E 01	-9.07980E 01
5.00000E 01	5.31239E-01	-5.49419E 00	-1.00005E 02	2.49600E 01	2.79449E 01	-9.04172E 01
6.00000E 01	4.47049E-01	-6.99290E 00	-1.05283E 02	2.58746E 01	2.82575E 01	-9.02609E 01
7.00000E 01	3.87739E-01	-8.22921E 00	-1.02396E 02	2.65091E 01	2.84679E 01	-9.01799E 01
8.00000E 01	3.43619E-01	-9.27844E 00	-1.00430E 02	2.70354E 01	2.86387E 01	-9.01318E 01
9.00000E 01	3.09902E-01	-1.01755E 01	-9.89894E 01	2.75487E 01	2.88020E 01	-9.01007E 01
1.00000E 02	2.83921E-01	-1.09360E 01	-9.78767E 01	2.81243E 01	2.89816E 01	-9.00793E 01
2.00000E 02	1.01425E-01	-1.98771E 01	-9.61141E 01	2.01696E 01	2.60940E 01	-9.00307E 01
3.00000E 02	7.88747E-02	-2.20612E 01	-9.31037E 01	2.36277E 01	2.74684E 01	-9.00104E 01
4.00000E 02	6.04551E-02	-2.43713E 01	-9.22471E 01	2.41635E 01	2.76632E 01	-9.00056E 01
5.00000E 02	4.87509E-02	-2.62403E 01	-9.17748E 01	2.43638E 01	2.77349E 01	-9.00036E 01
6.00000E 02	4.07848E-02	-2.77900E 01	-9.14698E 01	2.44628E 01	2.77701E 01	-9.00025E 01
7.00000E 02	3.50364E-02	-2.91096E 01	-9.12553E 01	2.45196E 01	2.77903E 01	-9.00018E 01
8.00000E 02	3.06998E-02	-3.02573E 01	-9.10959E 01	2.45554E 01	2.78029E 01	-9.00014E 01
9.00000E 02	2.73144E-02	-3.12722E 01	-9.09727E 01	2.45794E 01	2.78114E 01	-9.00011E 01
1.00000E 03	2.45992E-02	-3.21816E 01	-9.08745E 01	2.45963E 01	2.78174E 01	-9.00009E 01
2.00000E 03	1.23250E-02	-3.81843E 01	-9.04358E 01	2.46493E 01	2.78361E 01	-9.00002E 01
3.00000E 03	8.21973E-03	-4.17029E 01	-9.02905E 01	2.46589E 01	2.78395E 01	-9.00001E 01
4.00000E 03	6.16560E-03	-4.42005E 01	-9.02177E 01	2.46622E 01	2.78406E 01	-9.00001E 01
5.00000E 03	4.93277E-03	-4.61382E 01	-9.01742E 01	2.46638E 01	2.78412E 01	-9.00000E 01
6.00000E 03	4.11078E-03	-4.77215E 01	-9.01451E 01	2.46646E 01	2.78415E 01	-9.00000E 01
7.00000E 03	3.52359E-03	-4.90603E 01	-9.01244E 01	2.46651E 01	2.78417E 01	-9.00000E 01
8.00000E 03	3.08318E-03	-5.02200E 01	-9.01088E 01	2.46654E 01	2.78418E 01	-9.00000E 01
9.00000E 03	2.74063E-03	-5.12430E 01	-9.00967E 01	2.46657E 01	2.78419E 01	-9.00000E 01
1.00000E 04	2.46658E-03	-5.21581E 01	-9.00871E 01	2.46658E 01	2.78419E 01	-9.00000E 01
2.43640E-01	1.91684E 02	4.56517E 01	-8.98135E 01	4.67058E 01	3.33874E 01	-8.92348E 01

TABLE 4. Continued

OMEGA	/M/	/M/-DB	PHI	/M/P	/M/P-DB	PHI-P
1.21020E 00	3.84103E 01	3.16889E 01	-8.90679E 01	4.67894E 01	3.34029E 01	-8.92348E 01
2.43640E 00	1.93216E 01	2.57204E 01	-8.81378E 01	4.70546E 01	3.34520E 01	-8.92355E 01
4.07280E 00	9.90778E 00	1.99195E 01	-8.62984E 01	4.81822E 01	3.36577E 01	-8.92394E 01
7.30920E 00	6.91694E 00	1.67983E 01	-8.45476E 01	5.03329E 01	3.40370E 01	-8.92518E 01
9.74560E 00	5.58789E 00	1.49505E 01	-8.30780E 01	5.40646E 01	3.46583E 01	-8.92863E 01
1.21820E 01	5.01793E 00	1.40105E 01	-8.24340E 01	6.05999E 01	3.56494E 01	-8.93753E 01
1.46184E 01	4.99486E 00	1.39705E 01	-8.42032E 01	7.26453E 01	3.72241E 01	-8.94021E 01
1.70548E 01	5.51717E 00	1.48343E 01	-9.32164E 01	9.39466E 01	3.94576E 01	-9.01888E 01
1.94912E 01	5.74973E 00	1.51929E 01	-1.19288E 02	9.77842E 01	3.98054E 01	-9.16483E 01
2.19276E 01	3.81459E 00	1.16290E 01	-1.47424E 02	4.51501E 01	3.30932E 01	-9.40827E 01
2.43640E 01	2.22777E 00	6.95739E 00	-1.54187E 02	2.37191E 01	2.75020E 01	-9.48502E 01
4.87280E 01	5.44700E-01	-5.27685E 00	-1.10831E 02	2.48079E 01	2.78918E 01	-9.04474E 01
9.74560E 01	2.89893E-01	-1.07552E 01	-9.81357E 01	2.79675E 01	2.89331E 01	-9.00840E 01
1.46184E 02	2.68970E-01	-1.14059E 01	-9.57300E 01	3.91226E 01	3.18486E 01	-9.00393E 01
1.94912E 02	1.00522E-01	-1.99547E 01	-9.66996E 01	1.94593E 01	2.57825E 01	-9.00345E 01
2.43640E 02	9.34485E-02	-2.05886E 01	-9.40766E 01	2.27102E 01	2.71244E 01	-9.00168E 01
1.67769E 00	2.79435E 01	2.89247E 01	-8.87167E 01	4.68680E 01	3.34175E 01	-8.92350E 01
8.38845E 00	6.20329E 00	1.58524E 01	-8.38419E 01	5.17400E 01	3.42765E 01	-8.92631E 01
1.67769E 01	5.43532E 00	1.47045E 01	-9.15162E 01	9.11560E 01	3.91957E 01	-9.00904E 01
3.35538E 01	8.47953E-01	-1.43257E 00	-1.30640E 02	2.15969E 01	2.66878E 01	-9.14653E 01
5.03307E 01	5.27868E-01	-5.54949E 00	-1.09800E 02	2.49979E 01	2.79581E 01	-9.04098E 01
6.71076E 01	4.03039E-01	-7.89307E 00	-1.03111E 02	2.63421E 01	2.84130E 01	-9.01989E 01
8.38845E 01	3.29467E-01	-9.64377E 00	-9.98217E 01	2.72321E 01	2.87016E 01	-9.01182E 01
1.00661E 02	2.82435E-01	-1.09816E 01	-9.78117E 01	2.81665E 01	2.89947E 01	-9.00781E 01
1.17438E 02	2.53446E-01	-1.19223E 01	-9.64261E 01	2.95772E 01	2.94191E 01	-9.00549E 01
1.34215E 02	2.44201E-01	-1.12451E 01	-9.54978E 01	3.26247E 01	3.02709E 01	-9.00411E 01
1.50992E 02	3.09045E-01	-1.01996E 01	-9.71819E 01	4.62973E 01	3.33111E 01	-9.00478E 01
1.67769E 02	4.62744E-02	-2.66932E 01	-1.73228E 02	9.16547E-01	-7.56909E-01	-9.28738E 01
3.35538E 02	7.12907E-02	-2.29393E 01	-9.27262E 01	2.38937E 01	2.75657E 01	-9.00081E 01
6.71076E 02	3.65269E-02	-2.87477E 01	-9.13105E 01	2.45059E 01	2.77854E 01	-9.00020E 01
1.00661E 03	2.44385E-02	-3.22385E 01	-9.08687E 01	2.5973E 01	2.78177E 01	-9.00009E 01
1.34215E 03	1.83508E-02	-3.47269E 01	-9.06503E 01	2.46280E 01	2.78286E 01	-9.00005E 01
1.67769E 03	1.46886E-02	-3.66604E 01	-9.05198E 01	2.46420E 01	2.78335E 01	-9.00003E 01
1.97737E-01	2.36175E 02	4.74647E 01	-8.98487E 01	4.67046E 01	3.33872E 01	-8.92348E 01
9.88686E-01	4.72946E 01	3.39622E 01	-8.92434E 01	4.67596E 01	3.33974E 01	-8.92348E 01
1.97737E 00	2.37412E 01	2.75101E 01	-8.84878E 01	4.69331E 01	3.34296E 01	-8.92352E 01
3.95474E 00	1.20658E 01	2.16311E 01	-8.69862E 01	4.76553E 01	3.35622E 01	-8.92373E 01
5.93212E 00	8.27937E 00	1.83599E 01	-8.55194E 01	4.89684E 01	3.37983E 01	-8.92432E 01
7.90949E 00	6.49057E 00	1.62457E 01	-8.41471E 01	5.10737E 01	3.41640E 01	-8.92575E 01
9.88686E 00	5.53794E 00	1.48670E 01	-8.30098E 01	5.43500E 01	3.47040E 01	-8.92895E 01
1.18642E 01	5.06005E 00	1.40831E 01	-8.24330E 01	5.95146E 01	3.54925E 01	-8.93585E 01
1.38416E 01	4.94359E 00	1.38809E 01	-8.31900E 01	6.79470E 01	3.66434E 01	-8.95057E 01
1.58190E 01	5.18828E 00	1.43005E 01	-8.71953E 01	8.19754E 01	3.82737E 01	-8.98226E 01
1.77963E 01	5.73111E 00	1.51648E 01	-9.69839E 01	1.00746E 02	4.00645E 01	-9.05090E 01
1.97737E 01	5.62277E 00	1.49940E 01	-1.23276E 02	9.30043E 01	3.93701E 01	-9.19009E 01
3.95474E 01	6.77500E-01	-3.38181E 00	-1.19733E 02	2.32685E 01	2.73354E 01	-9.08274E 01
7.90949E 01	3.47132E-01	-9.19010E 00	-1.00582E 02	2.69895E 01	2.86239E 01	-9.01353E 01
1.18642E 02	2.52016E-01	-1.19714E 01	-9.63434E 01	2.97168E 01	2.94600E 01	-9.00537E 01
1.58190E 02	6.52422E-01	-3.70843E 00	-1.22404E 02	8.71371E 01	3.88041E 01	-9.02299E 01
1.97737E 02	1.01144E-01	-1.99012E 01	-9.63511E 01	1.98773E 01	2.59671E 01	-9.00323E 01
1.59992E 00	2.92886E 01	2.91340E 01	-8.87761E 01	4.68529E 01	3.34147E 01	-8.92350E 01
7.90949E 00	6.43321E 00	1.61686E 01	-8.40886E 01	5.11937E 01	3.41843E 01	-8.92584E 01
1.59992E 01	5.22917E 00	1.43687E 01	-8.78503E 01	8.36039E 01	3.84445E 01	-8.98656E 01
3.19984E 01	9.23181E-01	-6.94260E-01	-1.34545E 02	2.10632E 01	2.64705E 01	-9.17618E 01
4.79976E 01	5.52811E-01	-5.14847E 00	-1.11339E 02	2.47154E 01	2.78594E 01	-9.04663E 01
6.39964E 01	4.21072E-01	-7.51288E 00	-1.03974E 02	2.61494E 01	2.83492E 01	-9.02229E 01
7.99960E 01	3.43635E-01	-9.27806E 00	-1.00431E 02	2.70352E 01	2.86386E 01	-9.01319E 01

TABLE 4. Continued

OMEGA	/M/	/M/-DB	PHI	/M/P	/M/P-DB	PHI-P
9.59957E 01	2.93512E-01	-1.06475E 01	-9.82913E 01	2.78812E 01	2.89062E 01	-9.00870E 01
1.11994E 02	2.61071E-01	-1.16662E 01	-9.68262E 01	2.90268E 01	2.92560E 01	-9.00612E 01
1.27994E 02	2.44497E-01	-1.22345E 01	-9.57759E 01	3.11352E 01	2.98650E 01	-9.00453E 01
1.43993E 02	2.59742E-01	-1.17092E 01	-9.55129E 01	3.72280E 01	3.14174E 01	-9.00384E 01
1.59992E 02	8.01243E-01	-1.92472E 00	-1.67605E 02	2.75277E 01	2.87954E 01	-9.16290E 01
3.19984E 02	7.44499E-02	-2.25627E 01	-9.28779E 01	2.37927E 01	2.75289E 01	-9.00090E 01
6.39968E 02	3.82770E-02	-2.83412E 01	-9.13757E 01	2.44890E 01	2.77794E 01	-9.00022E 01
9.59952E 02	2.56193E-02	-3.18287E 01	-9.09113E 01	2.45902E 01	2.78152E 01	-9.00009E 01
1.27994E 03	1.92399E-02	-3.43160E 01	-9.06821E 01	2.46241E 01	2.78272E 01	-9.00005E 01
1.59992E 03	1.54011E-02	-3.62489E 01	-9.05451E 01	2.46395E 01	2.78326E 01	-9.00003E 01

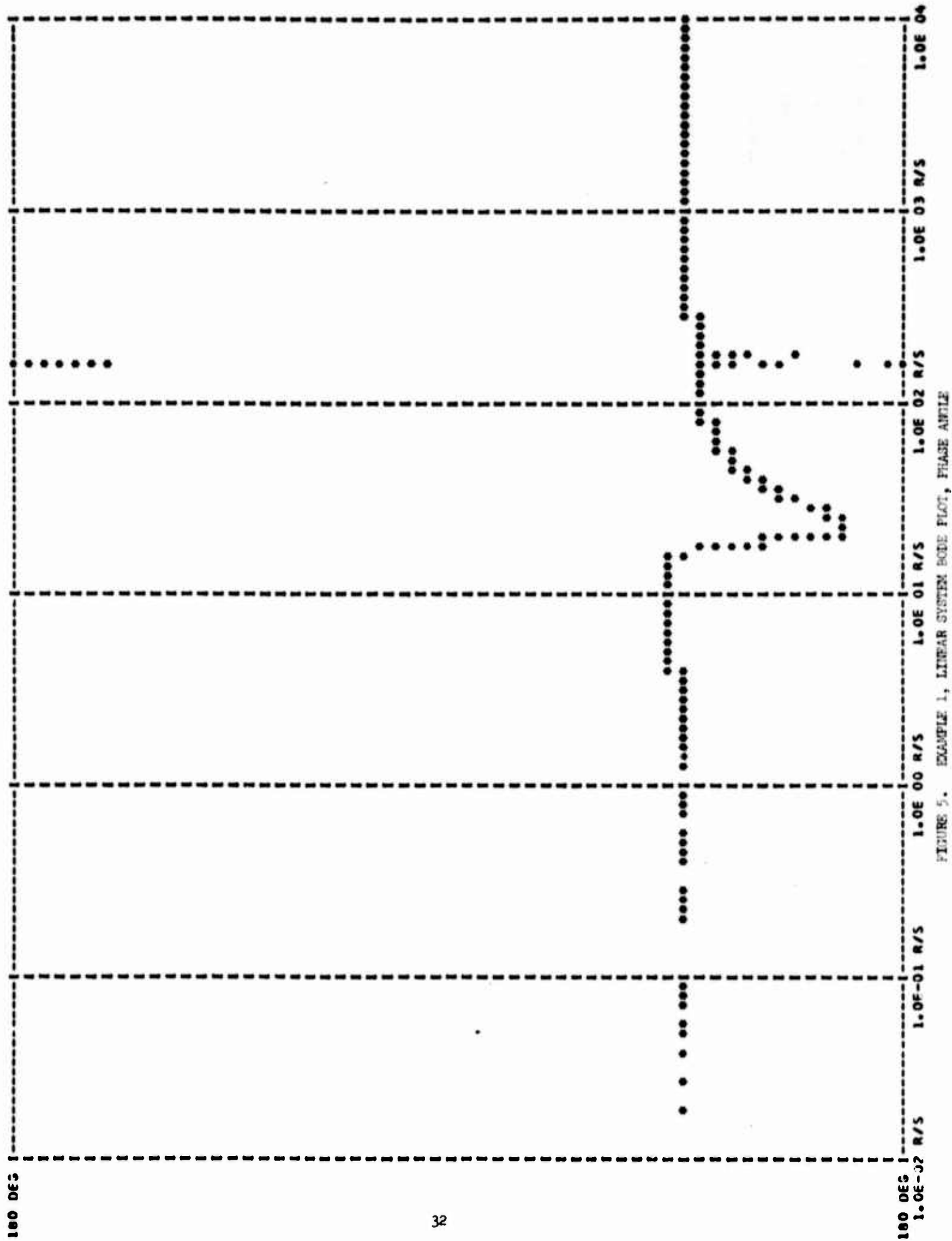


FIGURE 5. EXAMPLE 1, LINEAR SYSTEM BODE PLOT, PHASE ANGLE

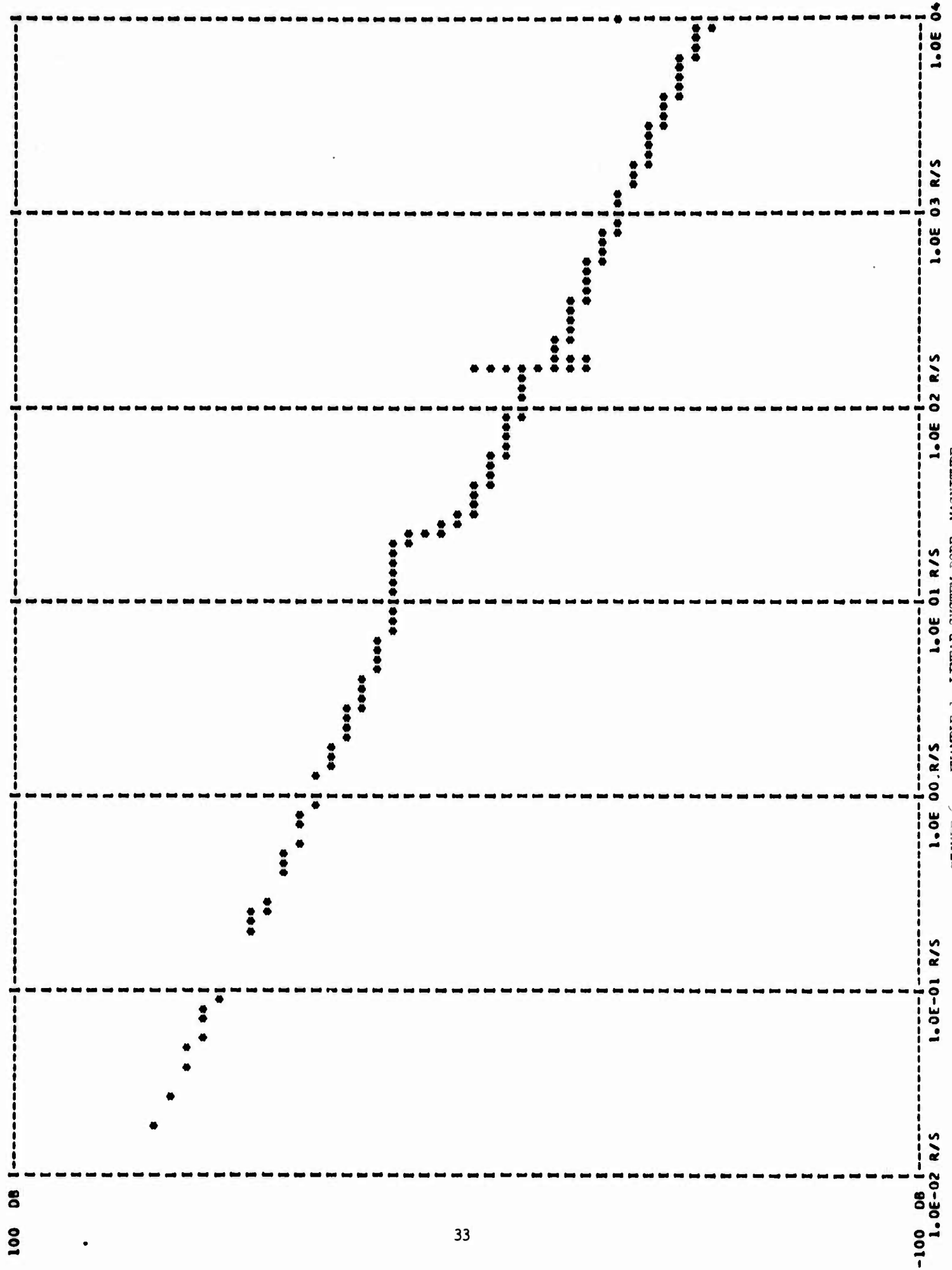


FIGURE 6. EXAMPLE 1, LINEAR SYSTEM BODE, MAGNITUDE

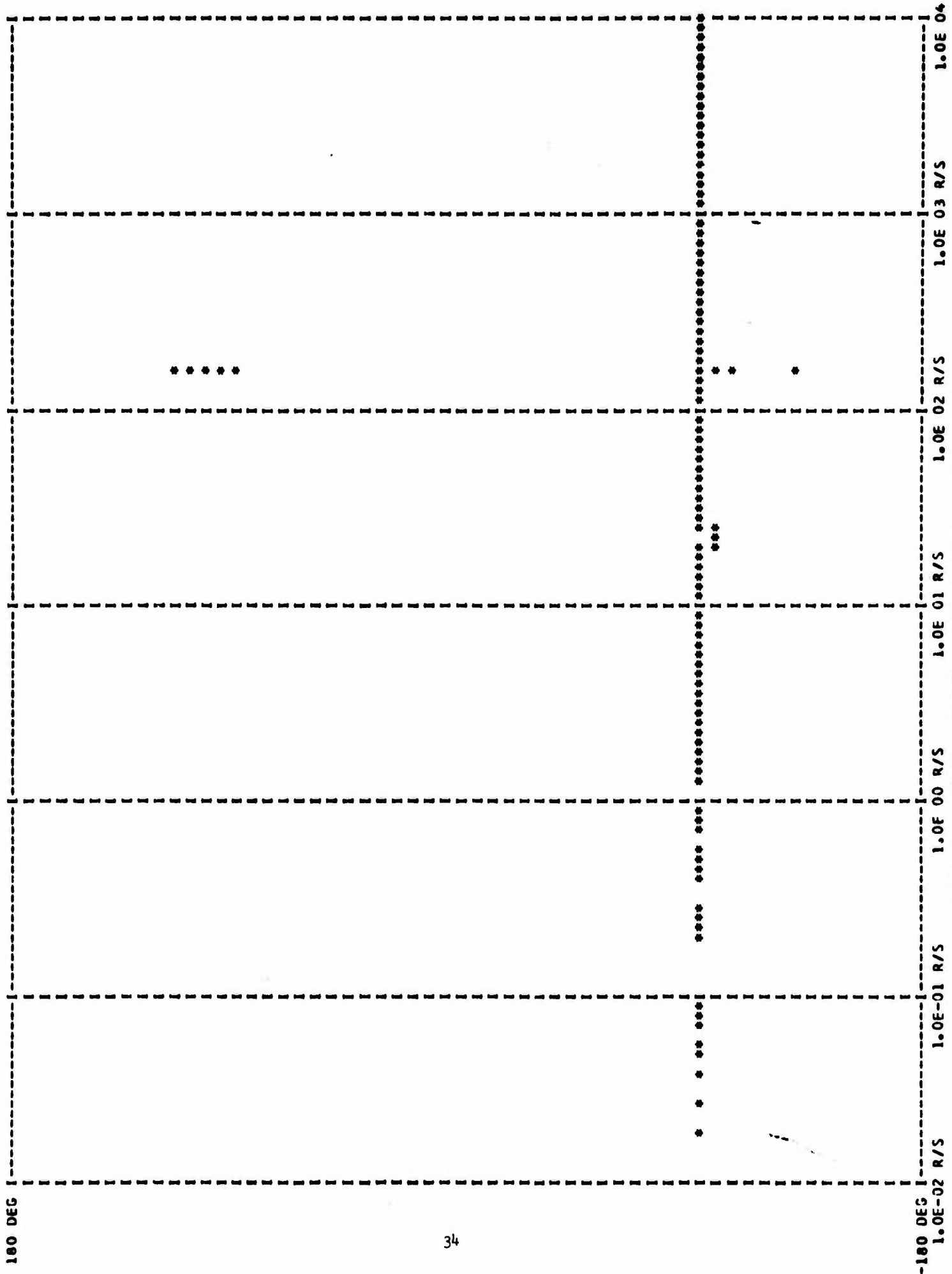


FIGURE 7. EXAMPLE 1. NONLINEAR SYSTEM, BODE PLOT, PHASE ANGLE

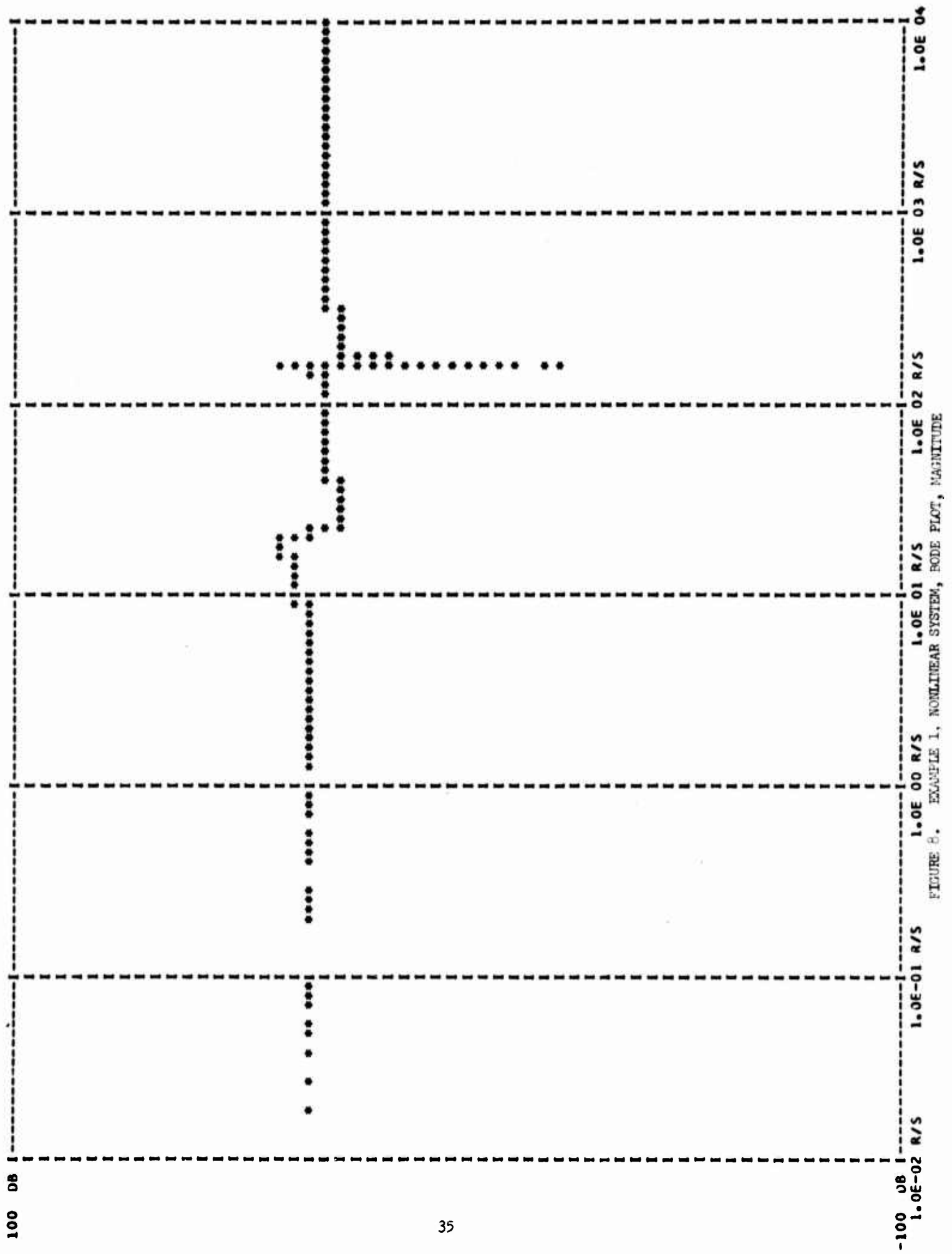


FIGURE 8. EXAMPLE 1, NONLINEAR SYSTEM, BODE PLOT, MAGNITUDE

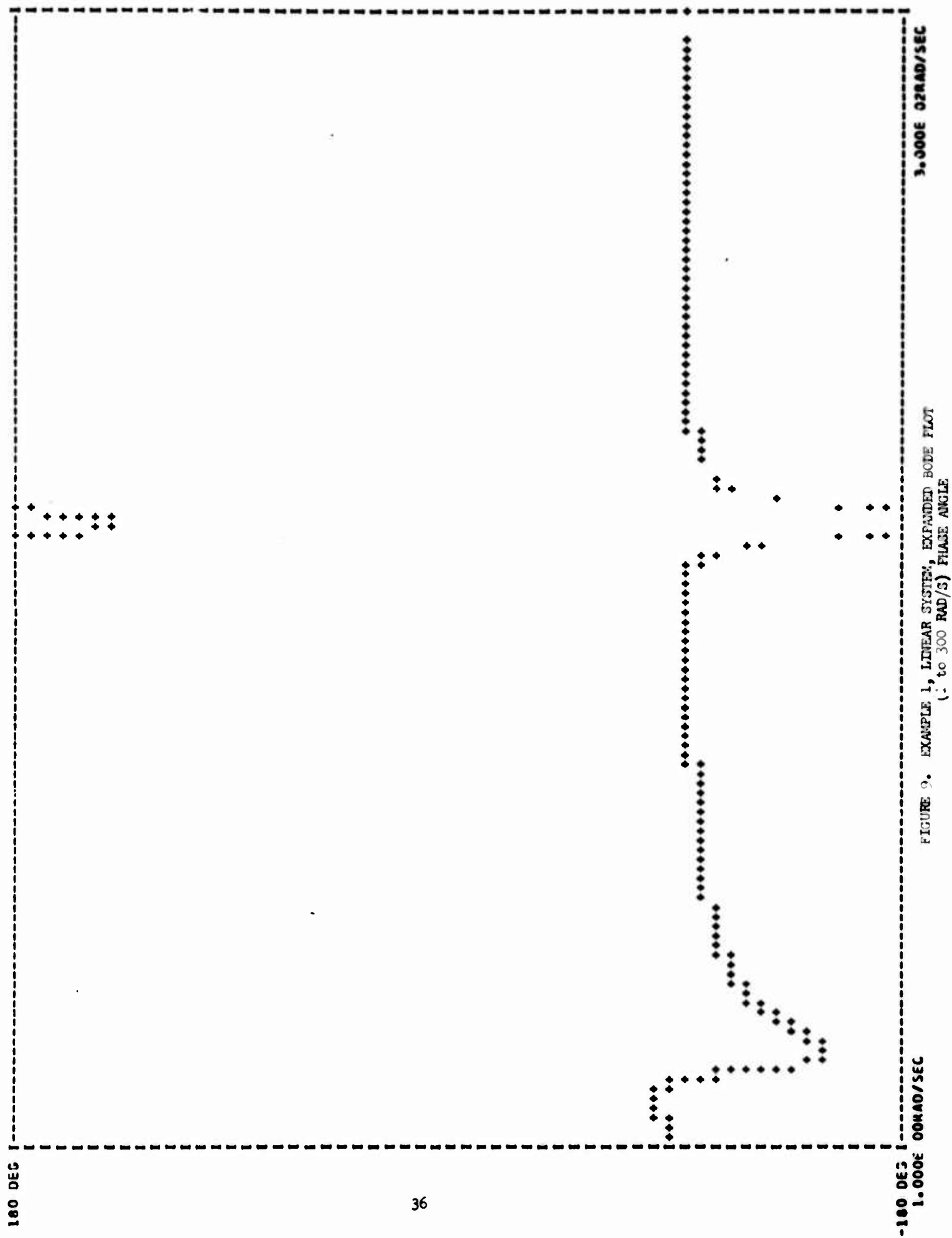
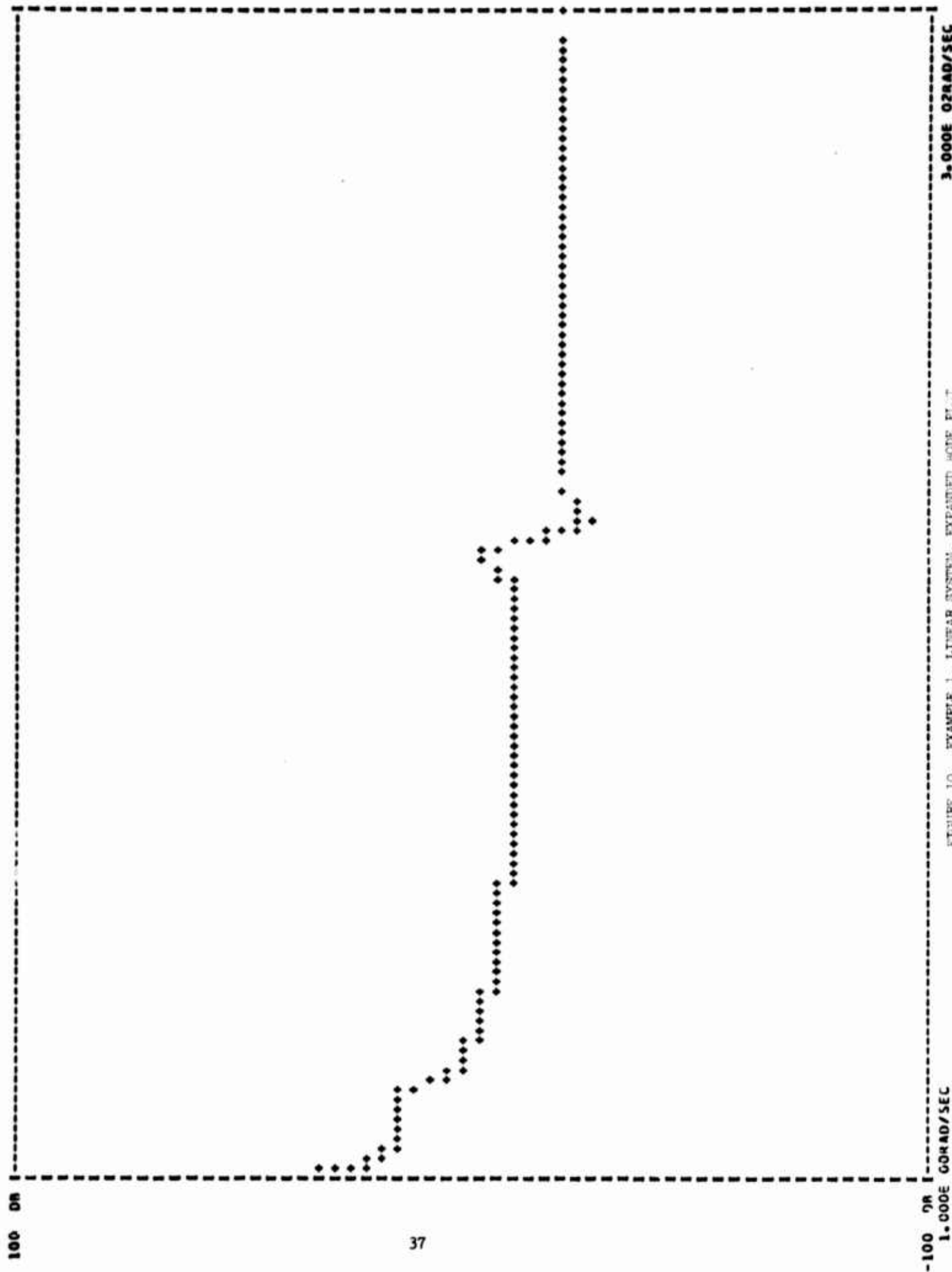


FIGURE 9. EXAMPLE 1, LINEAR SYSTEM, EXPANDED BODE PLOT
(0 to 300 RAD/S) PHASE ANGLE

100 DB



-100 DB

1.000E 00RAD/SEC

3.000E 02RAD/SEC

FIGURE 10. EXAMPLE 1, LINEAR SYSTEM, EXPANDED MODE PLOT
(1 TO 300 RAD/S) 1.000E 00

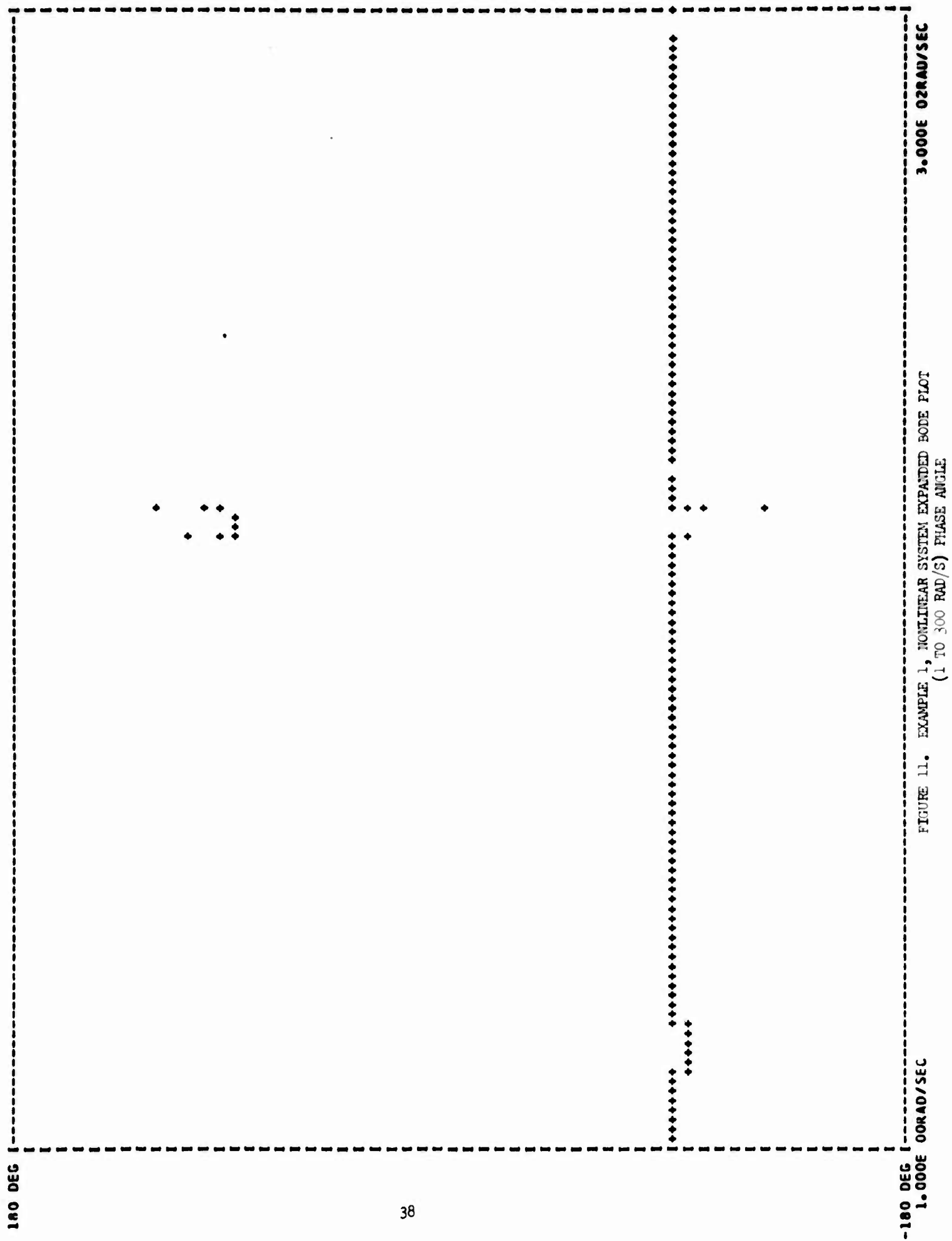


FIGURE 11. EXAMPLE 1, NONLINEAR SYSTEM EXPANDED BODE PLOT
(1 TO 300 RAD/S) PHASE ANGLE

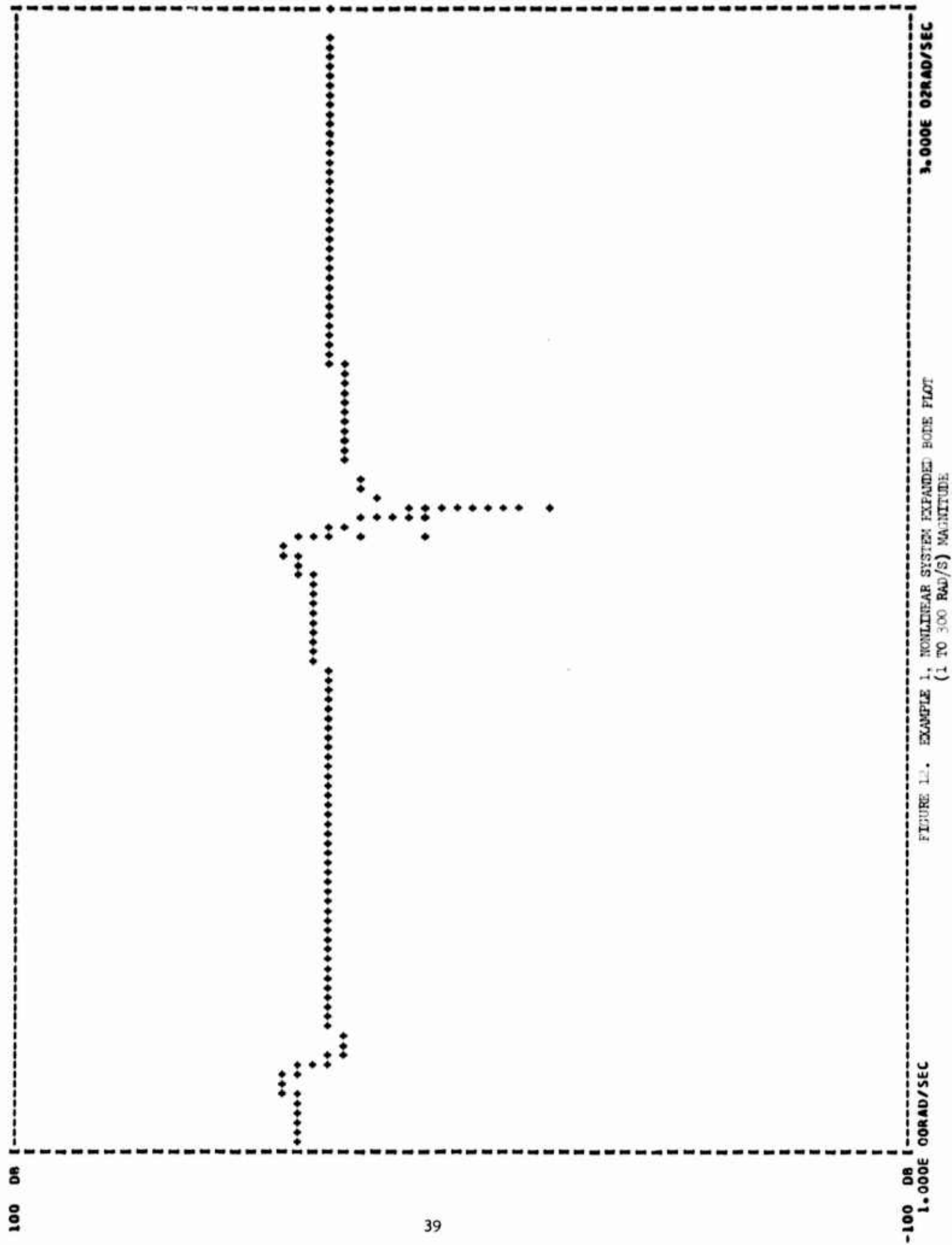


FIGURE 12. EXAMPLE 1. NONLINEAR SYSTEM EXPANDED BODE PLOT
(1 TO 300 RAD/S) MAGNITUDE

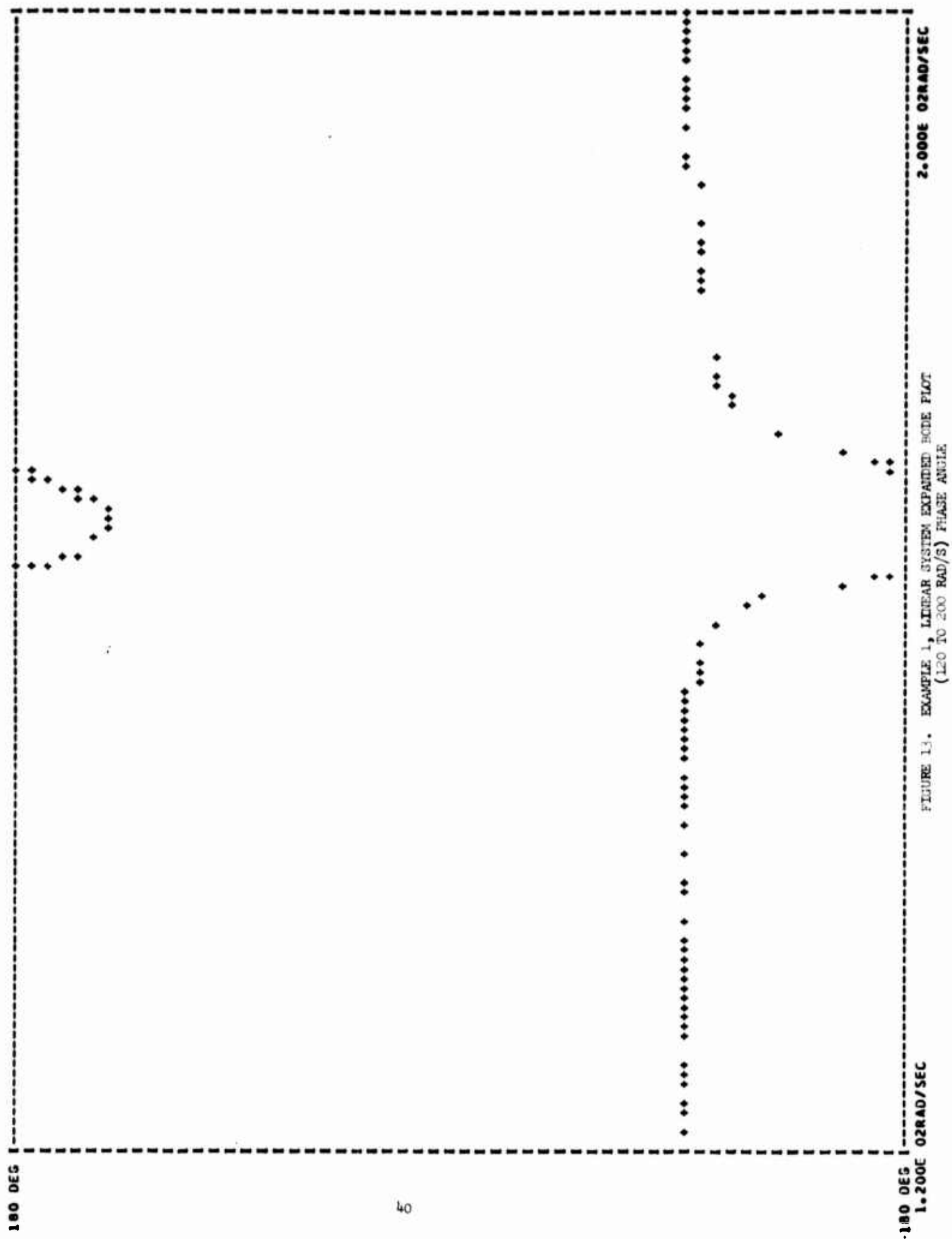


FIGURE 13. EXAMPLE 1, LINEAR SYSTEM EXPANDED BODE PLOT
(120 TO 200 RAD/S) PHASE ANGLE

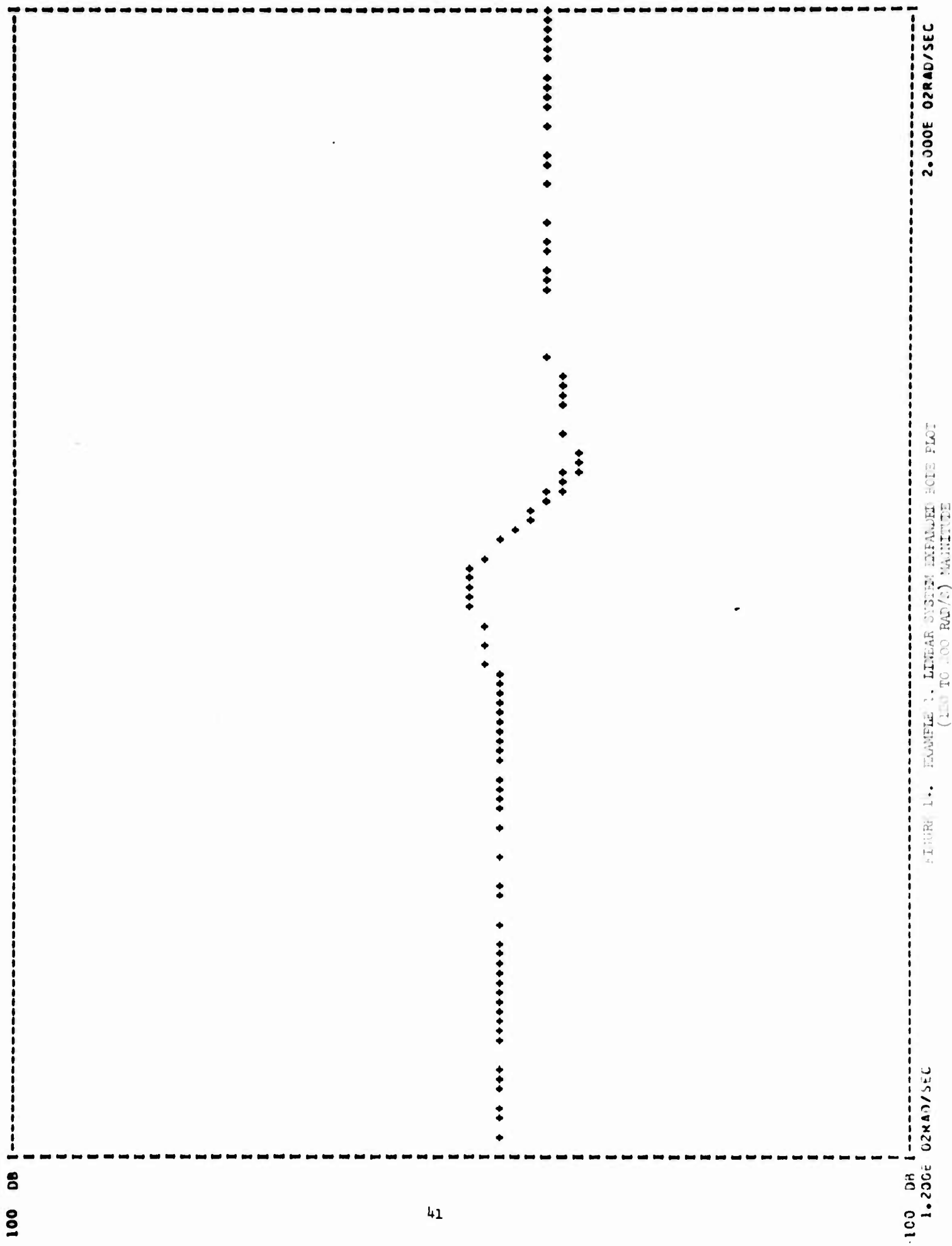


FIGURE 1. LINEAR SYSTEM EXPANDED BODE PLOT
(100 TO 200 RAD/S) MAGNITUDE

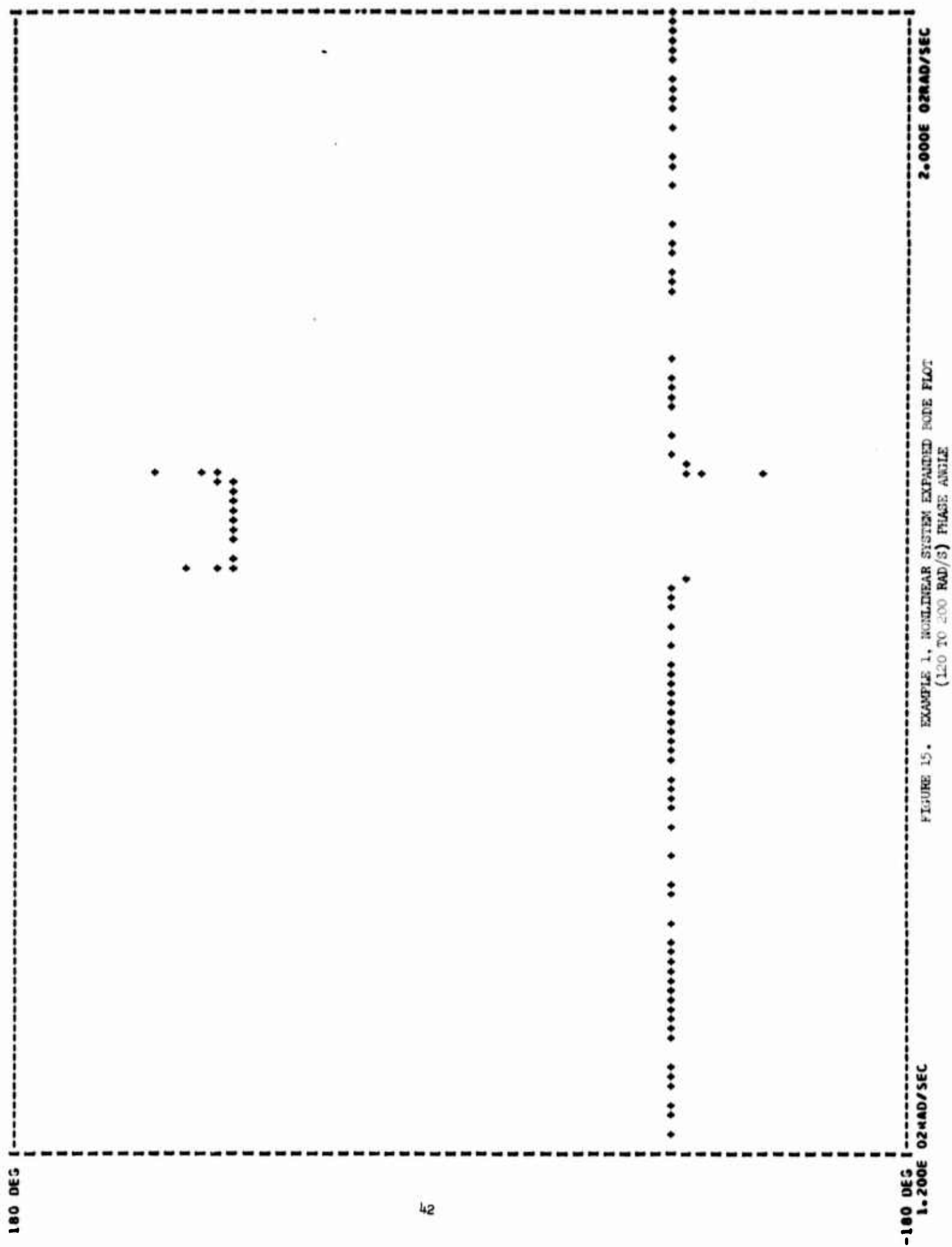
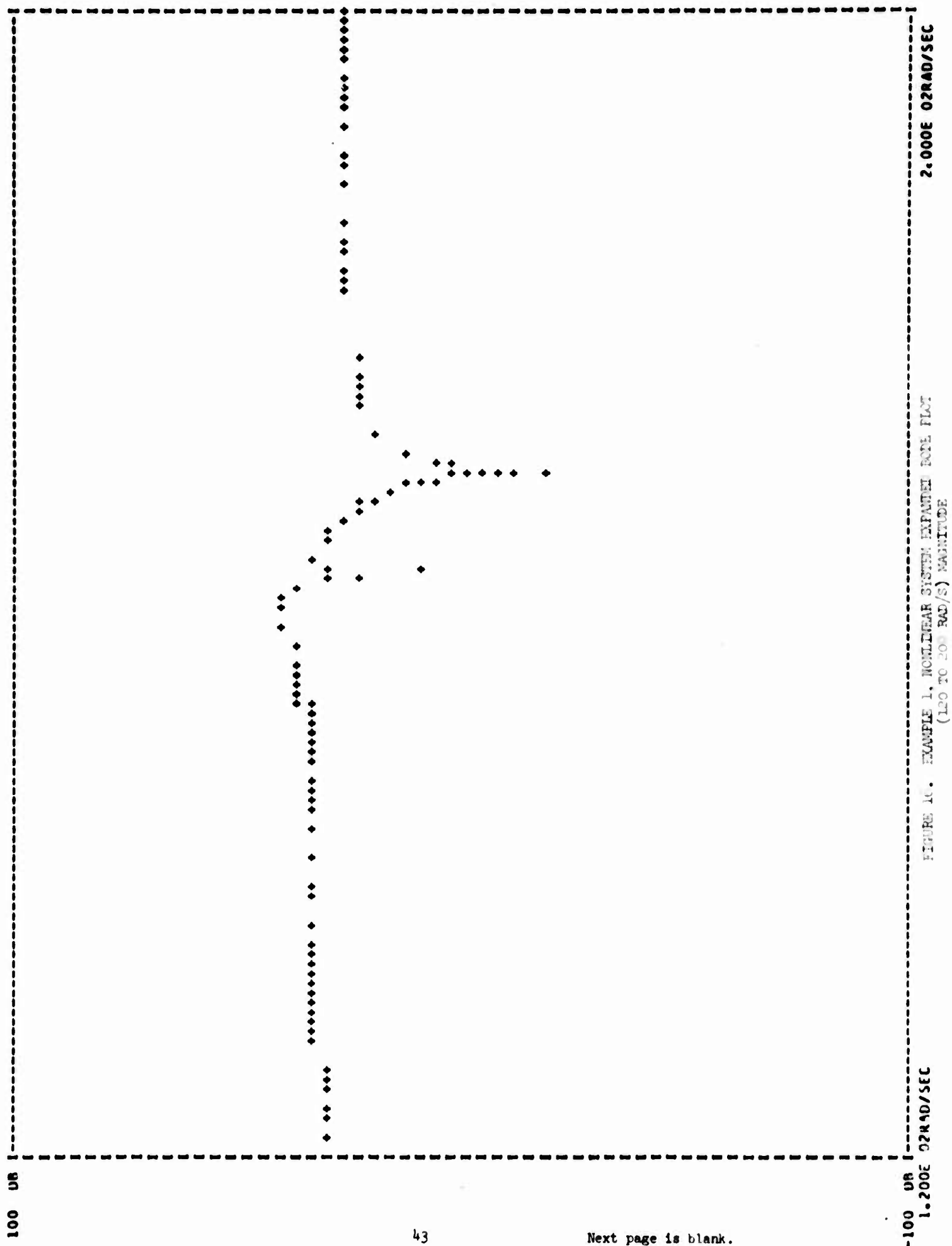


FIGURE 15. EXAMPLE 1. NONLINEAR SYSTEM EXPANDED BODE PLOT
(120 TO 200 RAD/S) PHASE ANGLE



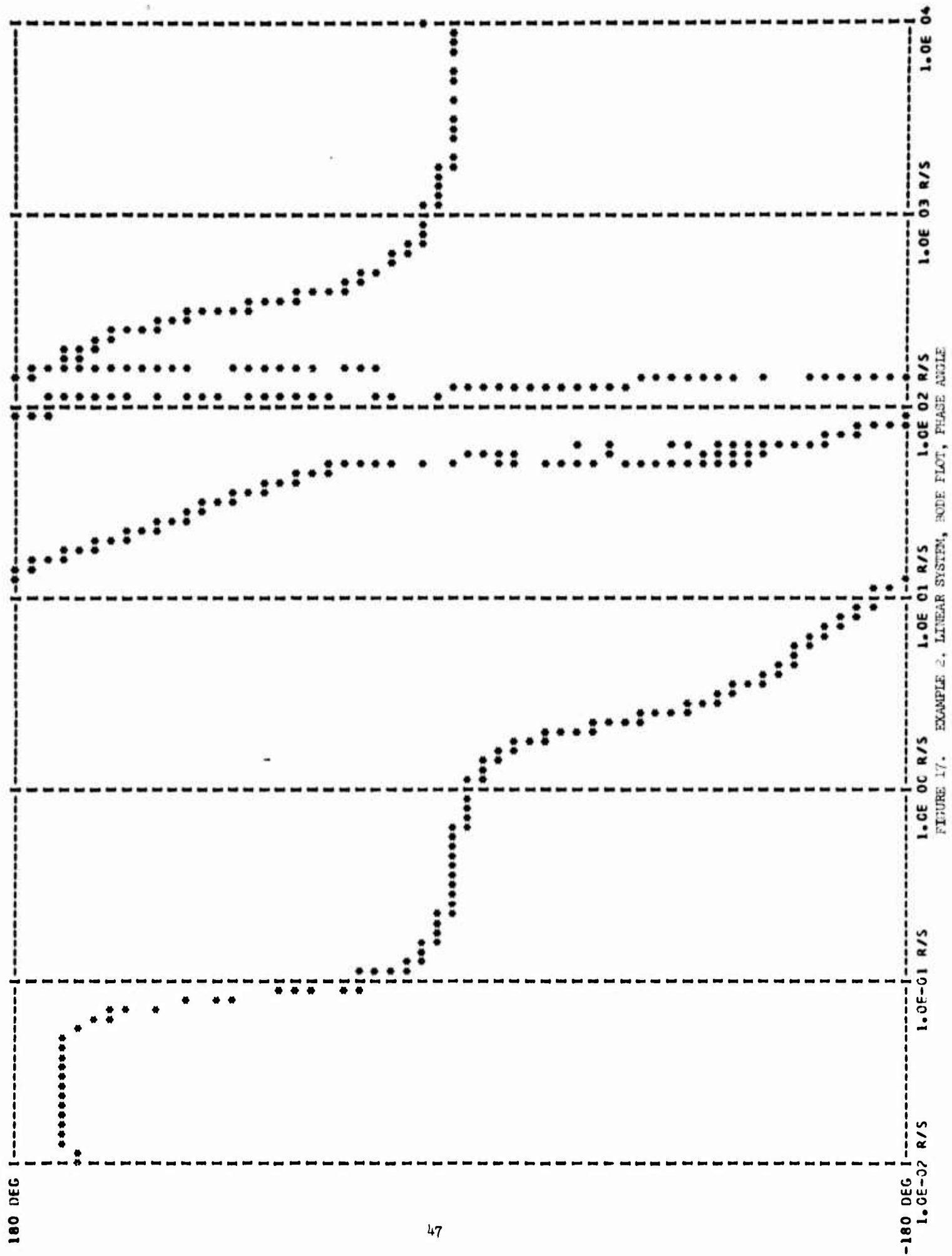
POLYNOMIAL MULTIPLICATION AND RCCT LOCUS

K-INITIAL=	DELTA=	0	PCLY.ADDED IN A(S)=	2	PCLY.ADDED IN B(S)=	1	PROB.NO.=	37
NUMBER OF POLY. IN GROUP 1 OF NUMERATOR=	NUMBER OF POLY. IN GROUP 2 OF NUMERATOR=	INCREMENT K=						
NUMBER OF POLY. IN GROUP 1 OF DENOMINATOR=								
C(1)=	.001500C000							
C(1)=	-.398190C000							
C(2)=	61.140700C000							
C(3)=	9194.180000C000							
C(4)=	45.774200C000							
C(5)=	-2.661700C000							
C(6)=	.081544C000							
C(7)=	.028599C000							
C(8)=	.0000673410							
C(9)=	.0000159750							
C(10)=	.0000000172							
C(11)=	.0000000026							
C(12)=	.0000000000							
C(13)=	.0000000000							
C(14)=	.0000000000							
C(15)=	.0000000000							
C(1)=	119300.000000C000							
C(1)=	37.500000C000							
C(1)=	12.500000C000							
C(2)=	62.800000C000							
C(3)=	1.0000000000							
C(2)=	2.0000000000							
C(1)=	180900.000000C000							
C(2)=	4000.000000C000							
C(1)=	3600.0000000000							
C(2)=	30.0000000000							
C(1)=	.0000000000							
C(2)=	.0000000000							
C(3)=	.0000000000							
C(4)=	.0000000000							
C(5)=	.0000000000							
C(6)=	.0000000000							
C(7)=	.0000000000							
C(8)=	.0000000000							
C(9)=	.0000000000							
C(10)=	.0000000000							
C(11)=	.0000000000							
C(12)=	.0000000000							
C(13)=	.0000000000							
C(14)=	.0000000000							
C(15)=	.0000000000							
C(1)=	119300.000000C000							
C(1)=	37.500000C000							
C(1)=	12.500000C000							
C(2)=	62.800000C000							
C(3)=	1.0000000000							
C(2)=	2.0000000000							
C(1)=	180900.000000C000							
C(2)=	4000.000000C000							
C(1)=	3600.0000000000							
C(2)=	30.0000000000							
C(1)=	.0000000000							
C(2)=	.0000000000							
C(3)=	.0000000000							
C(4)=	.0000000000							

TABLE 6. EQUIVALENT SYSTEM OPEN LOOP POLYNOMIALS
AND ROOTS, EXAMPLE 2.

COEFFICIENTS ARE GIVEN IN ASCENDING ORDER

THE COEFFICIENTS OF POLYNOMIAL A (ORDER = 20)									
-6.1198957E 16	9.2521124E 18	1.4379605E 21	3.5421717E 21	1.2905790E 21	5.9459794E 19				
9.0365506E 17	3.3735915E 16	3.9371677E 15	1.1777247E 14	2.9117079E 12	6.2795230E 10				
8.1833541E 08	1.1560946E 07	1.0646439E 05	9.3728004E 02	6.6668516E 00	3.3289968E-02				
1.8402498E-04	4.2351181E-07	1.7475004E-09							
THE ROOTS OF A									
-1.0694428E-02	+1 0.0000000E 00	-4.8391511E-01	+1 1.0342048E-14	4.0455430E-03	+1 0.0000000E 00				
-2.6281354E 06	+1 0.0000000E 00	-1.9524174E 01	+1 -1.2192916E 01	-1.9524174E 01	+1 1.2192916E 01				
1.8019734E 01	+1 -1.8023020E 01	1.8019734E 01	+1 1.8023020E 01	-4.4083790E-01	+1 5.8588405E 01				
-4.4083790E-01	+1 -5.8588405E 01	-1.0455319E 01	+1 1.0028177E 02	-1.0455319E 01	+1 -1.0028177E 02				
-4.5000000E 01	+1 0.0000000E 00	9.5034570E 00	+1 9.8992955E 01	9.5034570E 00	+1 -9.8992955E 01				
-3.2752621E 01	+1 1.9552017E 02	-3.2752621E 01	+1 -1.8552017E 02	-1.2000000E 02	+1 0.0000000E 00				
-1.4673431E 00	+1 1.6720316E 02	-1.4673431E 00	+1 -1.6720316E 02						
THE COEFFICIENTS OF POLYNOMIAL B (ORDER = 24)									
4.4350635E 18	2.7125958E 19	6.4489032E 20	1.5105939E 21	7.3257871E 20	3.1217410E 20				
4.6274362E 19	3.1607955E 18	1.2877217E 17	3.9926549E 15	1.0008113E 14	1.9949714E 12				
3.4274584E 10	4.7956796E 08	5.9506779E 06	5.8881896E 04	5.5416836E 02	3.8551691E 00				
2.8002674E-02	1.3359390E-04	7.3606134E-07	2.2365276E-09	8.7580072E-12	1.3306326E-14				
3.0508488E-17									
THE ROOTS OF B									
-5.0000000E-01	+1 0.0000000E 00	-1.3437380E-02	+1 8.4953454E-02	-1.3437380E-02	+1 -8.4953454E-02				
-8.5697546E-01	+1 2.2620353E 00	-8.5697546E-01	+1 -2.2620353E 00	-2.0000000E 01	+1 -8.0754456E-14				
-1.2500000E 01	+1 -8.7707630E-14	-4.6644787E-01	+1 5.1981022E 01	-4.6644787E-01	+1 -5.1981022E 01				
-4.3478261E 01	+1 4.4974263E 01	-4.3478261E 01	+1 -4.4974263E 01	-3.0000000E 01	+1 0.0000000E 00				
-6.0901678E-01	+1 6.0974886E 01	-6.0901678E-01	+1 -6.0974886E 01	-1.5634793E 00	+1 1.1108566E 02				
-1.5634793E 00	+1 -1.1108566E 02	-2.6156110E 00	+1 1.3130735E 02	-2.6156110E 00	+1 -1.3130735E 02				
-1.9225814E 00	+1 1.5928538E 02	-1.9225814E 00	+1 -1.5928538E 02	-3.1400000E 01	+1 1.8535922E 02				
-3.1400000E 01	+1 -1.8535922E 02	-1.0365000E 02	+1 3.2947940E 02	-1.0365000E 01	+1 -3.2947940E 02				



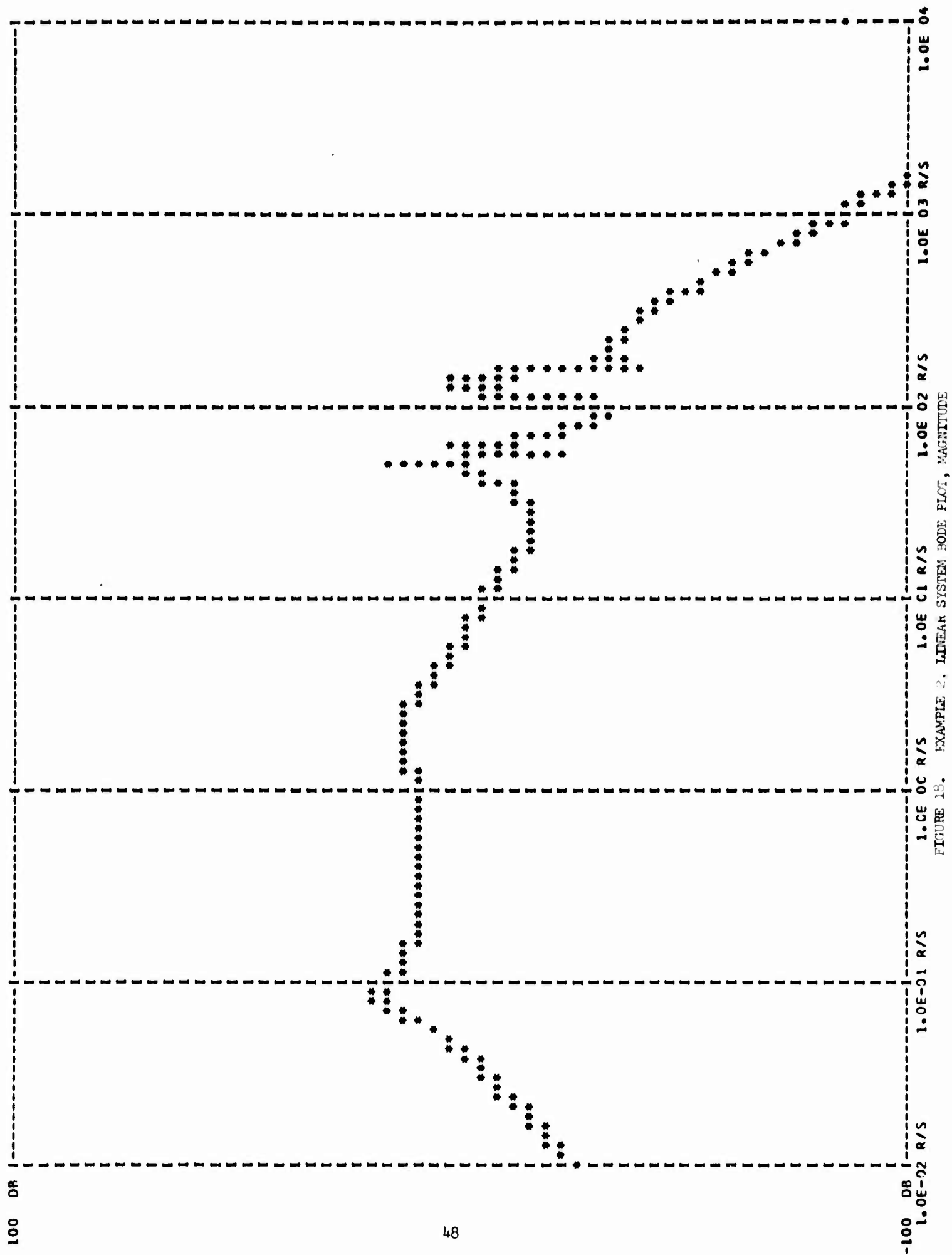


FIGURE 18. EXAMPLE 2. LINEAR SYSTEM BODE PLOT, MAGNITUDE

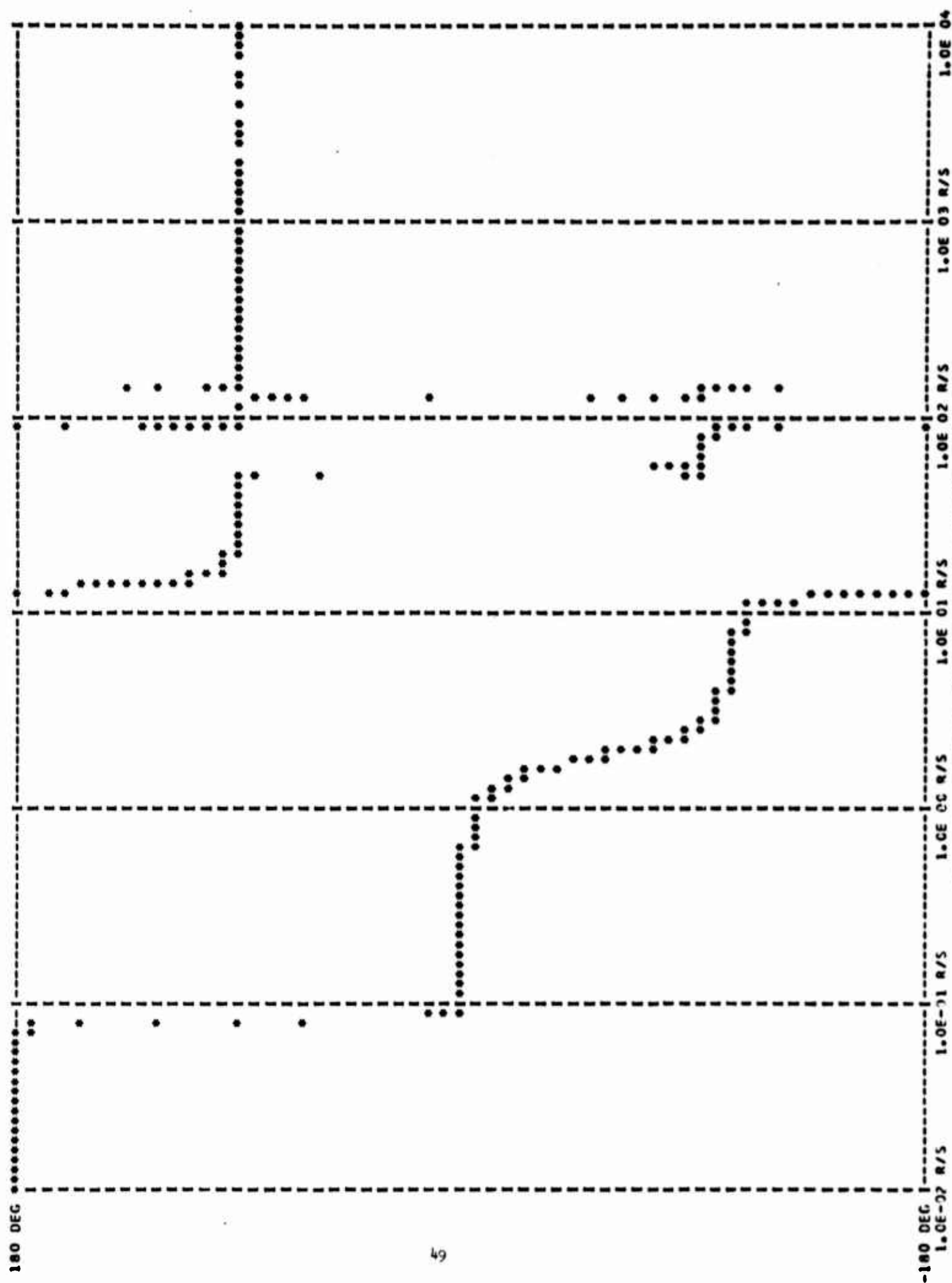


FIGURE 19. EXAMPLE 2. NONLINEAR SYSTEM, BODE PLOT, PHASE ANGLE

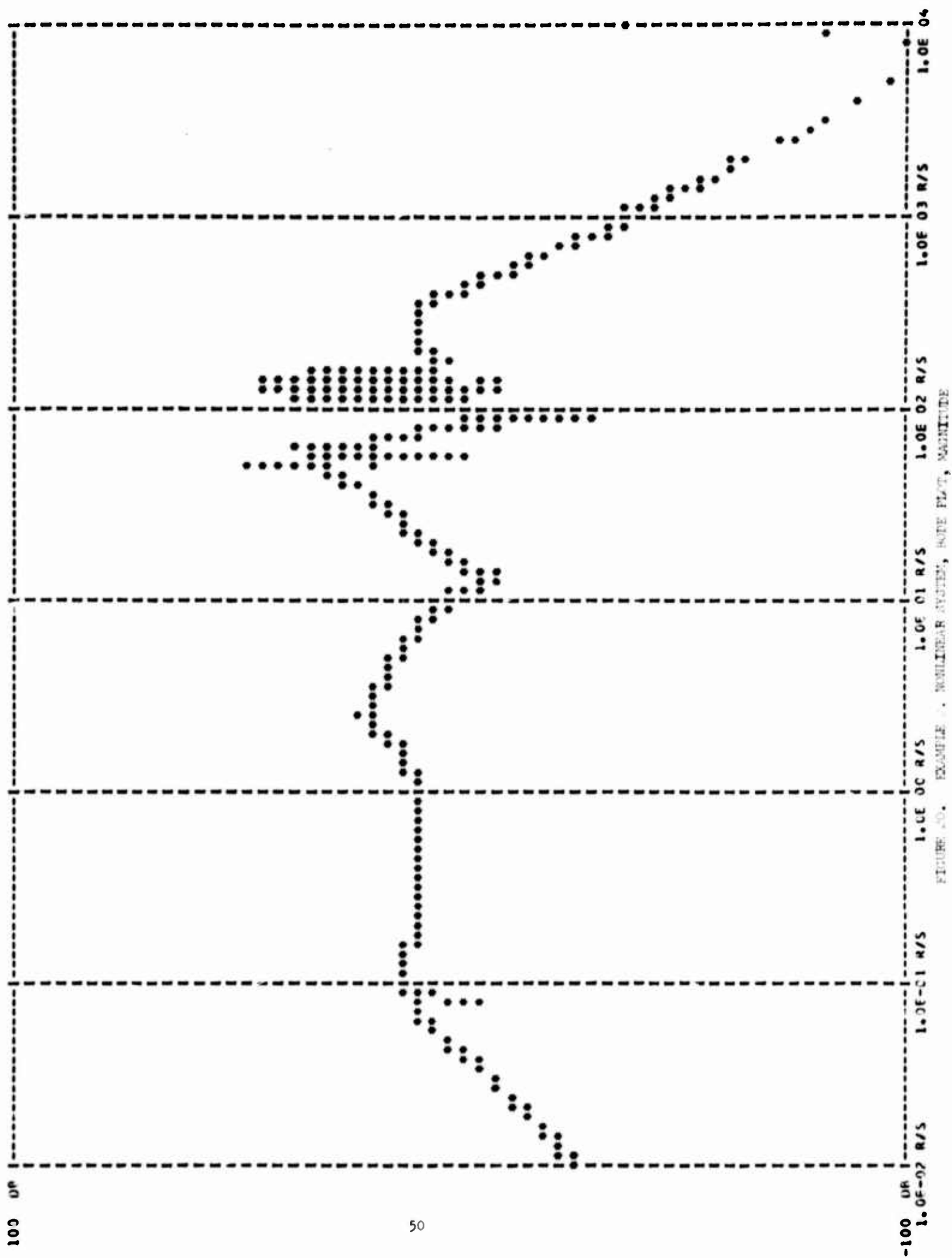


FIGURE 10. EXAMPLE 1. NONLINEAR ADJUSTED, NOISE PL/T, MAGNITUDE

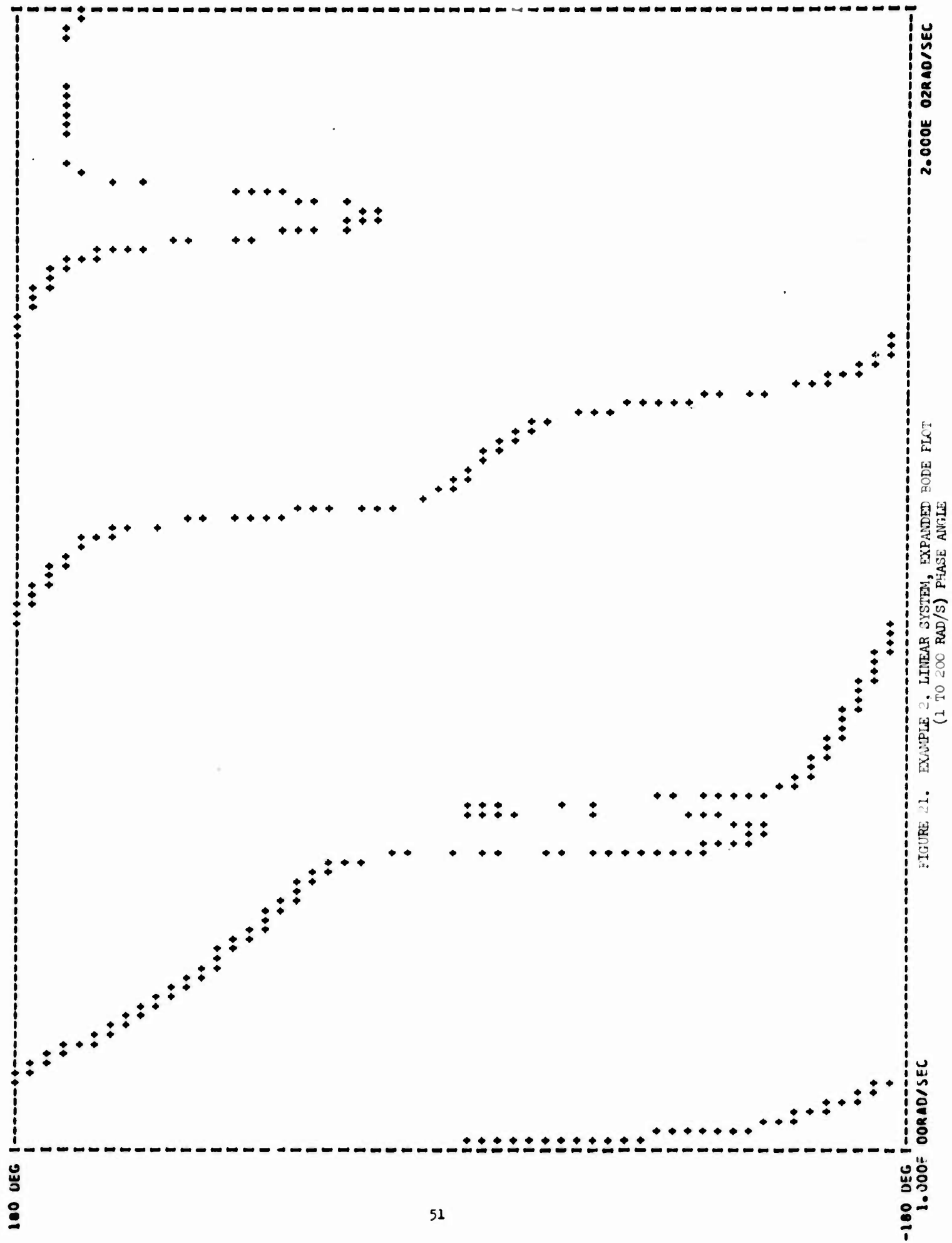
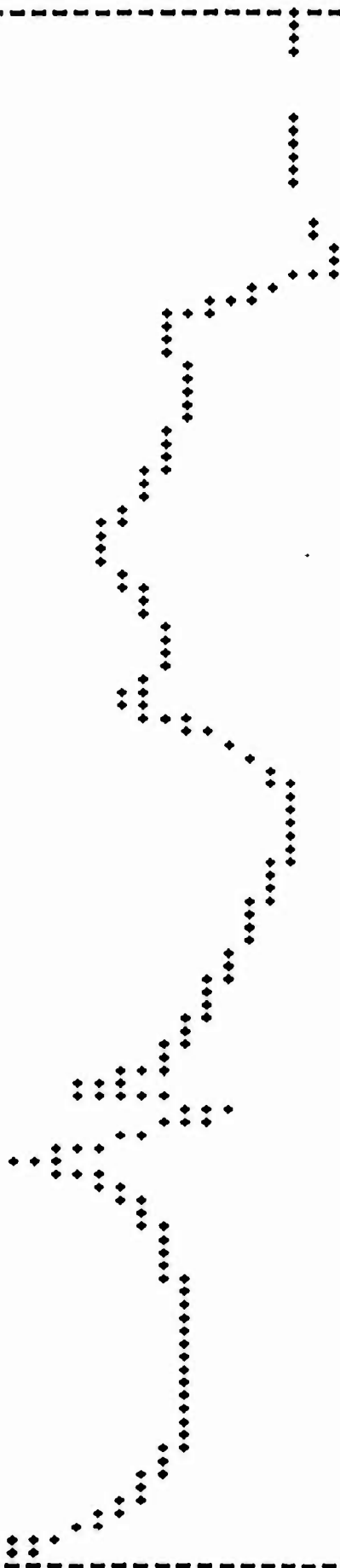


FIGURE 21. EXAMPLE 2. LINEAR SYSTEM, EXPANDED BODE PLOT
(1 TO 200 RAD/S) PHASE ANGLE

100 DB



-100 DB
1.000E RAD/SEC

FIGURE 12. EXAMPLE 1; LINEAR SYSTEM, EXPANDED BODE PLOT
(1 TO 200 RAD/SEC) MAGNITUDE

2.000E RAD/SEC

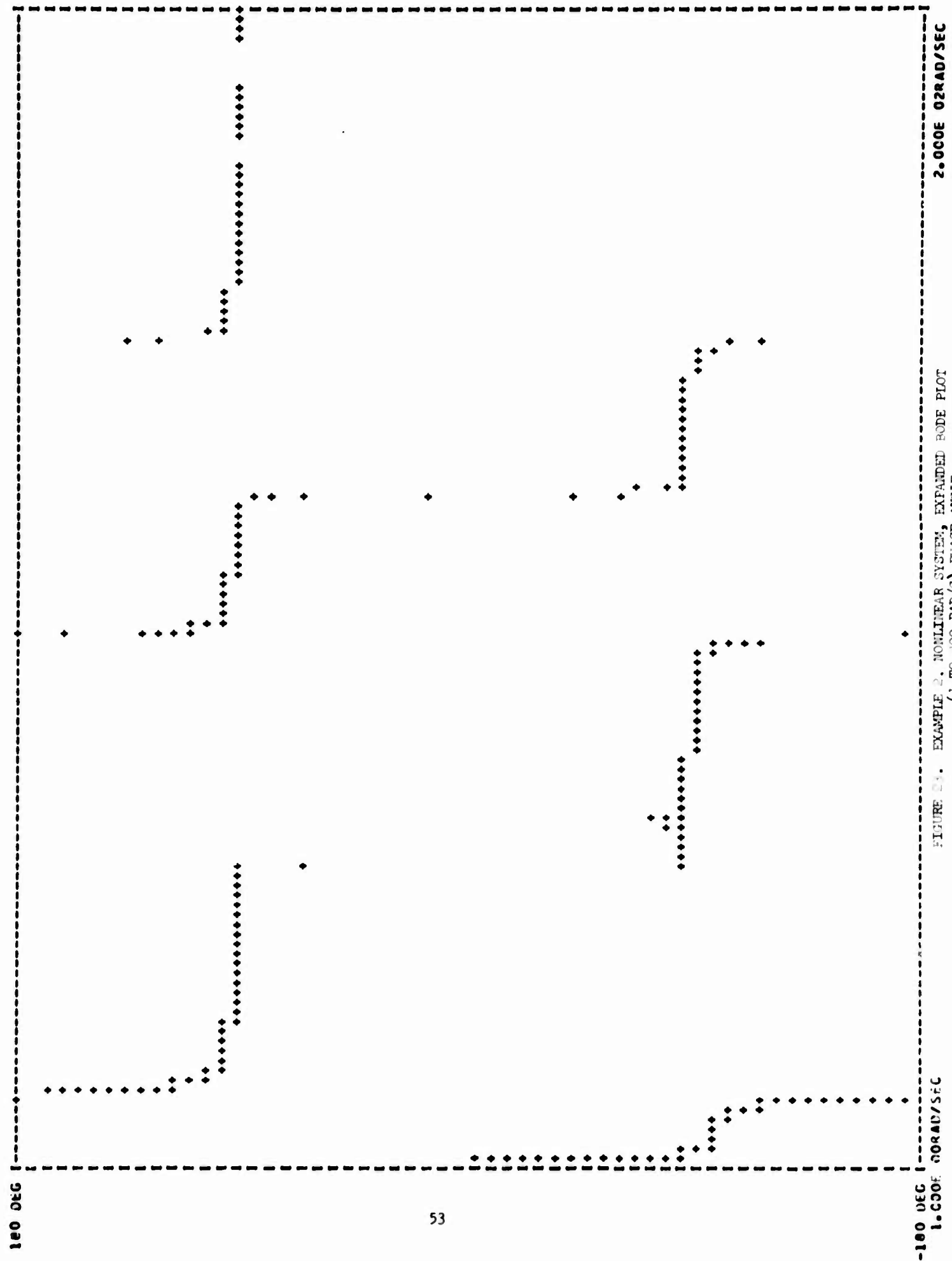


FIGURE 2.1. EXAMPLE 2. NONLINEAR SYSTEM, EXPANDED BODE PLOT
(1 TO 100 RAD/S) PHASE ANGLE

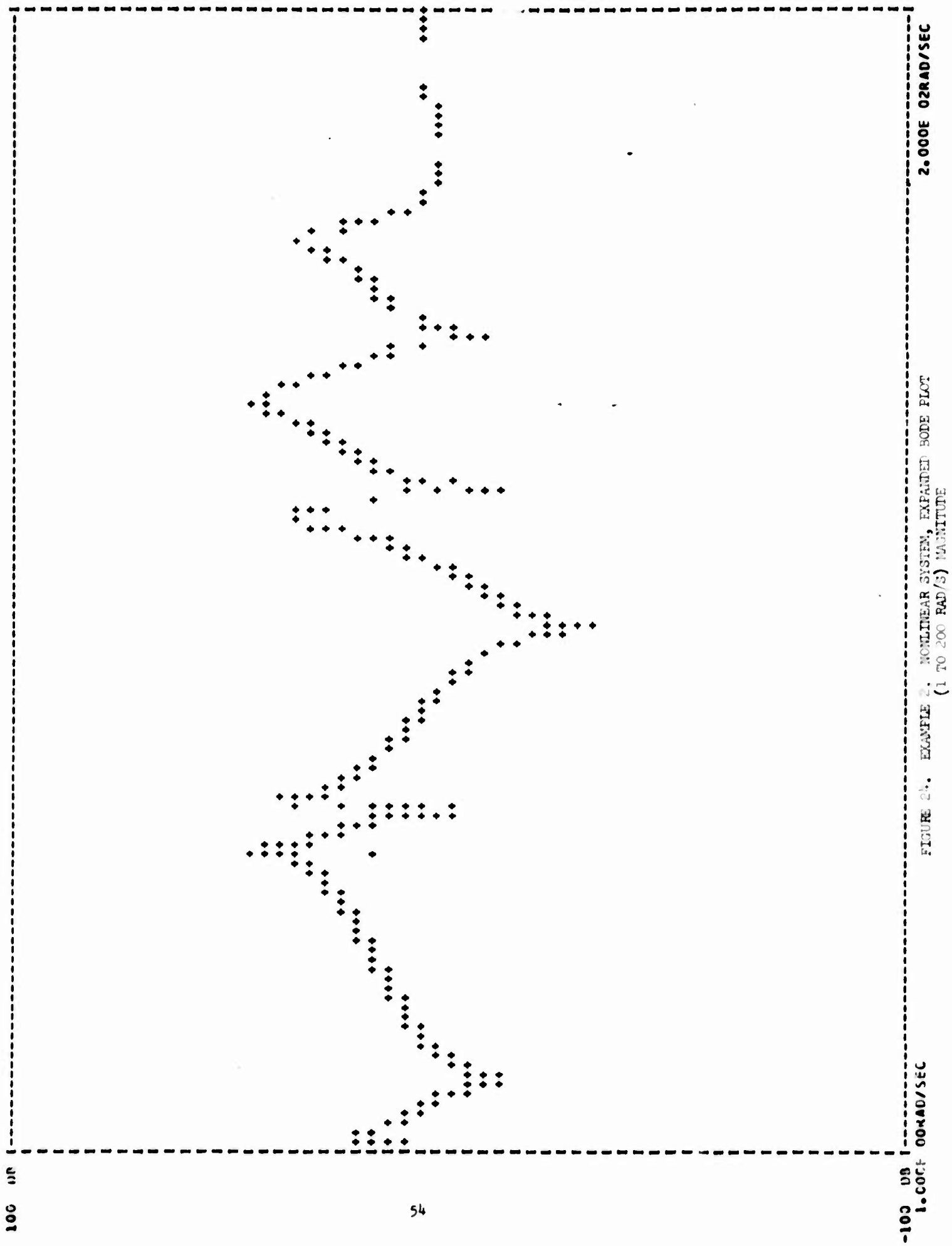


FIGURE 24. EXAMPLE 2. NONLINEAR SYSTEM, EXPANDED BODE PLOT
(1 TO 200 RAD/S) MAGNITUDE

REFERENCES

1. Burke, Harold H.; Payne, Robert L. Jr.; A Linear Closed Loop Systems Analysis Procedure Using Line Printer Plots of Characteristic Equation Root Loci, AMSAA Technical Memorandum No. 21, Army Materiel Systems Analysis Agency, Aberdeen Proving Ground, Md., November 1968, UNCLASSIFIED.
2. Burke, Harold H.; Andrese, Joseph A.; A Fortran IV Program to Compute the Inverse Laplace Transform and Plot the Response of a Linear System. AMSAA Technical Memorandum No. 60, Army Materiel Systems Analysis Agency, Aberdeen Proving Ground, Md., (In publication), UNCLASSIFIED.
3. Burke, Harold H.; Mathematical Model and Simulation Studies of a Pneumatic Valve-Actuator-Load Impedance Configuration, AMSAA Technical Report No. 25, Army Materiel Systems Analysis Agency, Aberdeen Proving Ground, Md., April 1969, UNCLASSIFIED.
4. Burke, Harold H.; Ogorzalek, M. J.; A Surface-To-Air Tactical Missile Guidance and Control Concept for the Advanced Forward Area Air Defense (AFAAD) Requirement, AMSAA Technical Memorandum No. 31, Army Materiel Systems Analysis Agency, Aberdeen Proving Ground, Md., September 1969 (CONFIDENTIAL).
5. Brown, James W.; A Preliminary Study of Shillelagh in a Combat, Optical Countermeasure Environment, AMSAA Technical Memorandum No. 62, Army Materiel Systems Analysis Agency, Aberdeen Proving Ground, Md., (In publication) (SECRET).
6. Sivaglian, B. D.; An Asymptotic Study of a Stationary (σ -s) Inventory Problem, Technical Report No. 8, June 1968, Themis Project, Systems Engineering Department, University of Florida, Gainesville, Fla.
7. Leon, Jesus, Herrero, Emilio E.; Determination of the Time-Domain Equivalents of Higher Order Laplace Transfer Functions By Analog Computation, Themis Project, Systems Research Center Industrial and Systems Engineering Department, University of Florida, Gainesville, Fla., November 1968.
8. Lefschetz, Solomon, Stability of Nonlinear Control Systems, Academic Press, New York, 1965.
9. Greensite, Arthur L.; Analysis and Design of Space Vehicle Flight Control Systems, Volume IV, Nonlinear Systems, NASA CR 823, July 1967.

APPENDIX A

LISTING OF SOURCE DECK

```

8      MAXT(10)MINS
        DO 10 J=1,100
        CALL RTLOCS
10     CONTINUE
        END
  
```

C
C
C
C

```

SUBROUTINE RTLOCS
  DIMENSION SAVE1(100,100),SAVE2(100,100),XA(2000),XB(2000)
  DIMENSION X(100),IGA(100),IGR(100),C(100),D(100),ANS(100),
1SAVE(100),ERA(100),AS(100),BS(100),A(100),R(100),ROGTR(100),
2ROOTI(100),ATK(100),CK(100),CKS(100)
  INTEGER COMENT(20),DENSE,ROTLCS,EXPND,FREQSR,STAR,STAR1,STAR2,
*      STAR3,ONE,ZERO,DOT,VEE,AYE,CASH,BLANK,Q,
*      POINT(130,100),SPOT(130,50)
  DATA ONE/1H1/,ZERO/1H0/,DOT/1H./,AYE/1HA/,VEE/1HV/,BLANK/1H /,
*      DASH/1H-/,Q/1H1/,STAR1/1H0/,STAR2/1H*/,STAR3/1H./
  LOGICAL PRNT
  EQUIVALENCE (SPOT(1,1),POINT(1,1)),(STAR1,ZERO),(STAR3,DOT),
*      (JACKIE,KJ,1),(JOANN,KK,J)
  COMMON/FREEK/A,B
  COMMON /INFO4/ Q,BLANK,DASH
  COMMON /INFO7/ ICOUNT
  COMMON SAVE1,SAVE2,POINT
  COMMON /INFO8/ROTLCS,EXPND,FREQSR
  COMMON /INFO9/ DENSE
  READ (5,105) ROTLCS,EXPND,FREQSR,DENSE
  READ(5,11) COMENT
  WRITE(6,13) COMENT
  IF(ROTLCS.NE.0.AND.EXPND.NE.0.AND.FREQSR.NE.0) GO TO 37
  IF(ROTLCS.EQ.0.AND.EXPND.EQ.0) WRITE(6,30)
  IF(FREQSR.EQ.0.AND.EXPND.EQ.0) WRITE(6,31)
  IF(FREQSR.EQ.0.AND.ROTLCS.EQ.0) WRITE(6,32)
  IF(ROTLCS.NE.0.AND.EXPND.NE.0) WRITE(6,33)
  IF(FREQSR.NE.0.AND.EXPND.NE.0) WRITE(6,34)
  IF(FREQSR.NE.0.AND.ROTLCS.NE.0) WRITE(6,35)
38  DO277 IOU=1,2000
      XA(IOU)=0.0
277  XB(IOU)=0.0
      DO 401 KK=1,100
      DO 401 KJ=1,130
401  POINT(KJ,KK)=BLANK
      DO 402 KK=1,130
      POINT(KK,51)=DASH
402  POINT(KK,50)=DASH
      DO 403 KK=1,100
      DO 403 KJ=34,125,13
403  POINT(KJ,KK)=Q
      DO 404 KJ=34,124
      DO 404 KK=7, 93,7
      IF(KK.EQ.49) KK=58
404  POINT(KJ,KK)=DASH
      DO 405 KK=7, 93,7
      IF(KK.EQ.49) KK=58
405  POINT(25,KK)=ONE
      DO 406 KK=26,28
406  POINT(KK,7)=ZERO
  
```

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58

DO 407 KK=26,27	59
407 POINT(KK,14)=ZERO	60
POINT(26,21)=ZERO	61
DO 408 KK=35,65,7	62
IF(KK.EQ.49) KK=58	63
408 POINT(24,KK)=DOT	64
POINT(25,42)=ZERO	65
POINT(25,58)=ZERO	66
POINT(26,42)=ONE	67
POINT(26,58)=ONE	68
POINT(26,79)=ZERO	69
POINT(26,86)=ZERO	70
POINT(27,86)=ZERO	71
DO 409 KK=26,28	72
409 POINT(KK,43)=ZERO	73
DO 420 KK=1,100	74
IF(KK.EQ.4) KK=98	75
420 POINT(26,KK)=Q	76
POINT(26,4)=VEE	77
POINT(26,97)=AYE	78
DO 7681 JACKIE=1,100	79
DO 7681 JOANN=1,100	80
SAVE1(JACKIE,JOANN)=0.0	81
7681 SAVE2(JACKIE,JOANN)=0.0	82
NO=1	83
ICOUNT=0	84
MI=0	85
JZ0=1	86
K1=1	87
III=1	88
140 FORMAT(10X,7HDELTA=,110,5X,19HPOLY.ADDED IN A(S)=,110,5X,19HPOLY.	89
*ADDED IN B(S)=,110,5X,9HPROR.NO.=,110)	90
12 READ(5,101) N,IA,IB,IPROB	91
IF(IP.NE.0) GO TO 10	92
IF(IA.NE.0) GO TO 10	93
IF(N.NE.0) GO TO 10	94
IF(IPROB.EQ.10000) GO TO 1016	95
101 FORMAT(7I10)	96
10 WRITE(6,1)	97
DO 5 I = 1,100	98
A(I) = 0.0	99
5 B(I) = 0.0	100
FORK1=0.0	101
FORK2 = 0.0	102
FORK3 = 0.0	103
WRITE(6,102)IPROB	104
102 FORMAT(1H1,9X,40HPOLYNOMIAL MULTIPLICATION AND ROOT LOCUS,44X,11	105
1HPROBLEM NO.,15//)	106
IF(N) 21, 20, 21	107
20 READ(5,103)Y,DY,YT	108
WRITE(6,140)N,IA,IB,IPROB	109
WRITE(6,141)Y,DY,YT	110
141 FORMAT(1H0,5X,10HK-INITIAL=F20.10,5X,12HINCREMENT K=,F20.10,5X,12H	111
*K-TERMINATE=,F20.10)	112
103 FORMAT(4F10.0)	113
GO TO 25	114
21 READ(5,104)(X(I),I=1,N)	115
WRITE(6,142)(I,X(I),I=1,N)	116
104 FORMAT(7E10.0)	117
25 READ(5,105)(IGA(I),I=1,IA)	118

WRITE(6,143) (I,IGA(I),I=1,IA)	119
143 FORMAT(10X,24HNUMBER OF POLY. IN GROUP, I F NUMERATOR=,I10)	120
142 FORMAT(10X,2HX(,13,2H)=,F20.10)	121
READ (5 ,105)(IGB(J), J= 1, IB)	122
WRITE(6,144) (J,IGB(J),J=1,IB)	123
144 FORMAT(10X,24HNUMBER CF POLY. IN GROUP, 13,16H OF DENOMINATOR=,I10	124
*)	125
105 FORMAT(7I10)	126
300 DO 15 I = 1,100	127
15 SAVF(I) = 0.0	128
ISDEG = 0	129
JJ = 1	130
200 READ (5 ,106)MDEG,(C(I), I=1, 6)	131
IF(MDEG.GT.6) GO TO 48	132
WRITE(6,145) (I,C(I),I=1,MDEG)	133
IF (MDEG - 6) 49, 49, 48	134
48 READ (5 ,104)(C(I), I = 7, MDEG)	135
WRITE(6,145) (I,C(I),I=1,MDEG)	136
106 FORMAT(110,6E10.0)	137
49 IF (IGA(JJ)-1) 50, 51, 50	138
51 DO 56 I= 1, MDEG	139
56 ANS(I) = C(I)	140
IADEG = MDEG	141
IF (ISDEG - MDEG) 52, 52, 53	142
53 INDEG = ISDEG	143
GO TO 68	144
52 INDEG = MDEG	145
GO TO 68	146
50 READ (5 ,106)NDEG, (D(I), I =1, 6)	147
IF(NDEG.GT.6) GO TO 54	148
WRITE(6,145) (I,D(I),I=1,NDEG)	149
IF (NDEG - 6) 55, 55, 54	150
54 READ (5 ,104)(D(I), I=7, NDEG)	151
WRITE(6,145) (I,D(I),I=1,NDEG)	152
55 IADEG = NDEG + MDEG -1	153
CALL POLMPY (C,MDEG,D,NDEG,ANS)	154
IGA (JJ) = IGA(JJ) -1	155
IF (IGA(JJ) -1) 65, 65, 64	156
64 DO 60 I = 1, IADEG	157
60 C(I) = ANS(I)	158
MDEG =IADEG	159
GO TO 50	160
65 IF (ISDEG - IADEG) 66,66,67	161
66 INDEG = IADEG	162
GO TO 68	163
67 INDEG = ISDEG	164
68 CALL POLADD (SAVE,ISDEG,ANS,IADEG,ERA)	165
145 FORMAT(10X,2HC(,13,2H)=,F20.10)	166
WRITE(6,1)	167
1 FORMAT(1H1//)	168
6800 IF (FRA(INDEG))6803, 6802, 6803	169
6802 INDEG = INDEG - 1	170
IF (INDEG) 6801, 6801, 6800	171
6801 INDEG = 1	172
6803 JJ = JJ + 1	173
DO 70 I = 1, INDEG	174
70 SAVE(I) = ERA(I)	175
ISDEG = INDEG	176
IA = IA -1	177
IF (IA) 201, 201, 200	178

201	IF (FORK1) 202, 202, 203	179
C	SAVE NUMERATOR.	180
202	DO 220 I = 1, ISDEG	181
220	A(I) = SAVE(I)	182
	IDA = ISDEG	183
	IA = IB	184
	FORK1 = 1.0	185
	DO 230 I=1, IA	186
230	IGA(I) = IGB(I)	187
C	START DENOMINATOR	188
	GO TO 300	189
C	SAVE DENOM.	190
203	DO 240 I = 1, ISDEG	191
240	B(I) = SAVE(I)	192
	IDB = ISDEG	193
	WRITE (6,109)	194
109	FORMAT (10X,41HCOEFFICIENTS ARE GIVEN IN ASCENDING ORDER/////)	195
339	IF (A(IDA)) 340, 341, 340	196
341	IDA = IDA - 1	197
	IF (IDA) 345, 345, 339	198
345	WRITE (6,120)	199
120	FORMAT (1H0,10X,20HPOLYNOMIAL A IS ZERO//)	200
	FORK2 = 1.0	201
	GO TO 410	202
340	IF (IDA - 2) 346, 347, 335	203
346	WRITE(6,121)A(1)	204
	STAR=STAR1	205
	PRNT=.TRUE.	206
121	FORMAT (1H0,10X,20HPOLYNOMIAL A IS A CONSTANT =,1P1E16.7//)	207
	GO TO 410	208
347	ROOT = - A(1) / A(2)	209
	WRITE (6,133)A(1), A(2)	210
133	FORMAT (10X,21HTHE COEFFICIENTS OF A/1P2E20.7)	211
	WRITE (6,122)ROOT	212
	STAR=STAR1	213
	PRNT=.TRUE.	214
	ANUMB1=ROOT	215
	ANUMB2=0.0	216
	CALL PLOTTER(SAVE1,SAVE2,ANUMB1,ANUMB2,POINT,XA,XB,MI,PRNT,STAR,III	217
	*,NO4	218
122	FORMAT (1H0,10X,23HROOT OF POLYNOMIAL A IS,1P1E16.7//)	219
	GO TO 410	220
C	WRITE POLYS	221
335	ID1A = IDA - 1	222
	WRITE (6,107)ID1A,(A(I),I=1,IDA)	223
	K = IDA	224
	DO 800 I = 1, IDA	225
	AS(I) = A(K)	226
800	K = K-1	227
	IDP2A=IDA *2	228
	ID2A= 2 *ID1A	229
	CALL MULLER (AS,ID1A,ROOTR,ROOTI)	230
	DO 805 I = 1, ID1A	231
	SAM = 100. * AMAX1(ABS(ROOTR(I)),ABS(ROOTI(I)))	232
	IF (SAM + ABS(ROOTR(I)).EQ. SAM) RCCTR(I) = 0.0	233
	IF (SAM + ABS(ROOTI(I)).EQ. SAM) RCOTI(I) = 0.0	234
805	CONTINUE	235
400	WRITE (6,111) (ROOTR(I),ROOTI(I),I=1, ID1A)	236
	CALL ERCHEK(ROOTI, ID1A)	237
	PRNT=.TRUE.	238

STAR=STAR1	239
DO 2 III=1, ID1A	240
ANUMB1=ROOTR(III)	241
ANUMB2=ABS(ROOTI(III))	242
CALL PLOTTER(SAVE1,SAVE2,ANUMB1,ANUMB2,POINT,XA,XB,MI,PRNT,STAR,III	243
*,NO)	244
2 CONTINUE	245
410 IF(B(IDB)) 411, 412, 411	246
412 IDB = IDB - 1	247
IF (IDB) 445, 445, 410	248
445 WRITE (6,123)	249
123 FORMAT (1H0,10X,20HPOLYNOMIAL B IS ZERO//)	250
IF (FORK2)12,450,12	251
450 FORK3 = 1.0	252
GO TO 698	253
411 IF (IDB - 2) 451, 452, 459	254
451 WRITE (6,124)B(IDB)	255
STAR=STAR2	256
PRNT=.FALSE.	257
STAR=STAR3	258
124 FORMAT (1H0,10X,28HPOLYNOMIAL B IS A CONSTANT =,1P1E16.7//)	259
GO TO 698	260
452 ROOT = -B(1) / B(2)	261
WRITE (6,134)B(1), B(2)	262
134 FORMAT (10X,21HTHE COEFFICIENTS OF B/1P2E20.7)	263
WRITE (6,125)ROOT	264
STAR=STAR2	265
PRNT=.TRUE.	266
ANUMB1=ROOT	267
ANUMB2=0.0	268
CALL PLOTTER(SAVE1,SAVE2,ANUMB1,ANUMB2,POINT,XA,XB,MI,PRNT,STAR,III	269
*,NO)	270
PRNT=.FALSE.	271
STAR=STAR3	272
125 FORMAT (1H0,10X,23HROOT OF POLYNOMIAL B IS,1P1E16.7//)	273
GO TO 698	274
107 FORMAT (10X,42HTHE COEFFICIENTS OF POLYNOMIAL A (ORDER = 13,1H)/ (275
11P6E20.7))	276
499 ID1B = IDB -1	277
WRITE (6,108)ID1B,(R(I),I=1,IDB)	278
108 FORMAT (////10X,42HTHE COEFFICIENTS OF POLYNOMIAL B (ORDER = 13,1H	279
1)/ (1P6E20.7))	280
K = IDB	281
DO 801 I = 1,ICR	282
BS(I) = B(K)	283
801 K = K-1	284
IDP2B= IDB + 2	285
ID2B = 2 * ID1B	286
CALL MULLER (BS, ID1B,ROOTR,ROOTI)	287
DO 806 I = 1, ID1B	288
SAM = 100. + AMAX1(ABS(ROOTR(I)),ABS(ROOTI(I)))	289
IF (SAM + ABS(ROOTR(I)).EQ. SAM) ROOTR(I)= 0.0	290
IF (SAM + ABS(ROOTI(I)).EQ. SAM) RCOTI(I)= 0.0	291
806 CONTINUE	292
500 WRITE (6,112)(ROOTR(I),ROOTI(I),I= 1,ID1B)	293
CALL ERCHEK(ROOTI,ID1B)	294
STAR=STAR2	295
PRNT=.TRUE.	296
DO 3 III=1, ID1B	297
ANUMB1=ROOTR(III)	298

ANUMB2=ABS(ROOT1(III))	299
CALL PLOTTER(SAVE1,SAVE2,ANUMB1,ANUMB2,POINT,XA,XB,MI,PRNT,STAR,III	300
*,NO)	301
3 CONTINUE	302
PRNT=.FALSE.	303
STAR=STAR3	304
111 FORMAT (1H0,11X,14HTHE ROOTS OF A/ (1P1E20.7,6H +I ,1P1E14.7,1P1	305
1E20.7,6H +I ,1P1E14.7,1P1E20.7,6H +I ,1P1E14.7))	306
112 FORMAT (1H0,11X,14HTHE ROOTS OF B/ (1P1E20.7,6H +I ,1P1E14.7,1P1	307
1E20.7,6H +I ,1P1E14.7,1P1E20.7,6H +I ,1P1E14.7))	308
698 IF (FORK2)12,699,12	309
699 IF (FORK3)12,699,12	310
6991 WRITE (6 ,102)IPRJB	311
MSHEET = 5	312
C START K CALCULATIONS	313
IF (N) 702,702,533	314
533 DO 550 I= 1, N	315
DO 541 J= 1, IDA	316
541 ATK(J) = X(I) * A(J)	317
C COMPUTE ROOTS OF K * A + B	318
IDC= MAX0(IDA, IDB)	319
CALL POLADD (ATK,IDA,R,IDB,CK)	320
IDS = IDC	321
554 IF (CK(IDS))555, 557, 555	322
557 IDS = IDS - 1	323
IF (IDS) 558,558, 554	324
558 WRITE (6 ,129)X(I)	325
129 FORMAT (1H0,10X,35HPOLYNOMIAL K*A + B IS ZERO FOR K =,1P1E16.7//)	326
GO TO 550	327
555 IF (IDS - 2) 559, 560, 561	328
559 WRITE (6 ,130)CK(IDS), X(I)	329
130 FORMAT (1H0,10X,35HPOLYNOMIAL K*A + B IS A CONSTANT = ,1P1E15.7,10	330
1H FOR K = ,1P1E14.7//)	331
GO TO 550	332
560 ROOT = -CK(1) / CK(2)	333
WRITE (6 ,131)ROOT, X(I)	334
131 FORMAT (1H0,10X,18HROOT OF K*A + B = ,1P1E15.7,10H FOR K = ,1P1E1	335
14.7//)	336
GO TO 550	337
561 K = IDS	338
DO 803 J = 1,IDS	339
CKS(J) = CK(K)	340
803 K = K - 1	341
ID1C = IDS - 1	342
IDP2C = IDS * 2	343
ID2C = 2 * ID1C	344
CALL MULLER (CKS,ID1C,ROOTR,ROOTI)	345
DO 807 J = 1,ID1C	346
SAM = 100. * AMAX1(ABS(ROOTR(J)),ABS(ROOTI(J)))	347
IF (SAM + ABS(ROOTR(J)).EQ.SAM) RCOTR(J) = 0.0	348
IF (SAM + ABS(ROOTI(J)).EQ.SAM) RCCTI(J) = 0.0	349
807 CONTINUE	350
WRITE (6,808)ID1C,X(I),(CK(J),J=1,IDS)	351
808 FORMAT (///10X,48HTHE COEFFICIENTS OF POLYNOMIAL K*A + B (ORDER =	352
113,7H) K = 1P1E16.7/(1P6E20.7))	353
545 WRITE (6,115)(ROOTR(J),ROOTI(J),J=1,ID1C)	354
CALL ERCHEK(ROOTI,ID1C)	355
CALL SAVER(ROOTR,ROOTI,ID1C,SAVE1,SAVE2,J20,K1)	356
115 FORMAT (1H0,9X,16HROOTS OF K*A + B/(1P1E20.7,6H + I ,1P1E14.7,1P1	357
1E20.7,6H + I ,1P1E14.7,1P1E20.7,6H + I ,1P1E14.7))	358

5452	MSHEET = MSHEET - 1	359
	IF (MSHEET) 546, 546, 550	360
546	WRITE (6,102)IPROB	361
	MSHEET = 5	362
550	CONTINUE	363
	GO TO 12	364
702	DO 705 J = 1, IDA	365
705	ATK(J) = Y * A(J)	366
C	COMPUTE ROOTS OF $K * A + B$	367
	IDC = MAX0(IDA, IDR)	368
	CALL POLADD (ATK, IDA, R, IDB, CK)	369
	IDS = IDC	370
754	IF (CK(IDS)) 755, 757, 755	371
757	IDS = IDS - 1	372
	IF (IDS) 758, 758, 754	373
758	WRITE (6,129)Y	374
	GO TO 711	375
755	IF (IDS - 2) 759, 760, 761	376
759	WRITE (6,130)CK(IDS), Y	377
	GO TO 711	378
760	ROOT = -CK(1) / CK(2)	379
	WRITE (6,131)ROOT, Y	380
	GO TO 711	381
761	K = IDS	382
	DO 804 I = 1, IDS	383
	CKS(I) = CK(K)	384
804	K = K - 1	385
	ID1C = IDS - 1	386
	IDP2C = IDS * 2	387
	ID2C = 2 * ID1C	388
	CALL MULLER (CKS, ID1C, ROOTR, ROOTI)	389
	DO 809 I = 1, ID1C	390
	SAM = 100. * AMAX1(ABS(ROOTR(I)), ABS(ROOTI(I)))	391
	IF (SAM + ABS(ROOTR(I)).EQ. SAM) ROOTR(I) = 0.0	392
	IF (SAM + ABS(ROOTI(I)).EQ. SAM) ROOTI(I) = 0.0	393
809	CONTINUE	394
	WRITE (6,808)ID1C,Y,(CK(I),I=1,IDS)	395
	WRITE (6,115) (ROOTR(J),ROOTI(J),J=1,ID1C)	396
	CALL ERCHER(ROOTI, ID1C)	397
	CALL SAVER(ROOTR, ROOTI, ID1C, SAVE1, SAVE2, JZ0, K1)	398
711	Y = Y + DY	399
	IF (Y - YT) 712, 712, 12	400
712	MSHEET = MSHEET - 1	401
	IF (MSHEET) 713, 713, 702	402
713	WRITE (6,102)IPROB	403
	MSHEET = 5	404
	GO TO 702	405
37	WRITE(6,36)	406
	GO TO 38	407
1016	CALL PLOTTER(SAVE1,SAVE2,ANUMB1,ANUMB2,POINT,XA,XB,MI,PRNT,STAR,III	408
	*,NO)	409
1018	IF(EXPND.EQ.0) GO TO 1017	410
	READ(5,22)I	411
22	FORMAT(I10)	412
	IF(I.EQ.0) GO TO 1017	413
	CALL EXPAND(L,XA,XB)	414
1017	IF(FREQSK.EQ.0) RETURN	415
	CALL FREQRS(XA,XB)	416
11	FORMAT(20A4)	417
13	FORMAT(//////////,20X,20A4)	418

30	FORMAT(////20X,44HONLY THE FREQUENCY RESPONSE HAS BEEN PLOTTED)	419
31	FORMAT(////20X,45HONLY THE ROOT LOCUS LOG PLOT HAS BEEN PLOTTED)	420
32	FORMAT(////20X,39HONLY THE LINEAR EXPAND HAS BEEN PLOTTED)	421
33	FORMAT(////20X,59HTHE ROOT LOCUS LOG PLOT AND LINEAR EXPAND HAVE B	422
	*EEN PLOTTED)	423
34	FORMAT(////20X,62HTHE LINEAR EXPAND AND THE FREQUENCY RESPONSE HAV	424
	*E BEEN PLOTTED)	425
35	FORMAT(////20X,68HTHE ROOT LOCUS LOG PLOT AND THE FREQUENCY RESPON	426
	*SE HAVE BEEN PLOTTED)	427
36	FORMAT(////20X,28HEVERYTHING HAS BEEN PLOTTED)	428
	RETURN	429
	END	430
C		431
C		432
C		433
C		434
	SUBROUTINE POLMPY (A,N,B,M,C)	435
	DIMENSION A(1),B(1),C(1)	436
	K = M+N	437
	DO 5 I=1,K	438
5	C(I) = 0.0	439
	DO 10 I=1,N	440
	L = I-1	441
	DO 10 J=1,M	442
	L = L+1	443
10	C(L) = C(L)+A(I)*B(J)	444
	RETURN	445
	END	446
C		447
C		448
C		449
C		450
	SUBROUTINE POLADD (A,N,B,M,C)	451
	DIMENSION A(1),B(1),C(1)	452
	IF (N-M) 1,1,2	453
1	NK = N	454
	GO TO 5	455
2	NK = M	456
5	DO 10 I=1,NK	457
10	C(I) = A(I)+B(I)	458
	NK = NK+1	459
	IF (N-M) 11,25,15	460
11	DO 20 I=NK,M	461
20	C(I) = B(I)	462
25	RETURN	463
15	DO 30 I=NK,N	464
30	C(I) = A(I)	465
	RETURN	466
	END	467
C		468
C		469
C		470
C		471
	SUBROUTINE MULLER(COE,N1,ROOTR,ROOTI)	472
	DIMENSION COE(1),ROOTR(1),ROOTI(1)	473
	N2=N1+1	474
	N4=0	475
	I=N1+1	476
19	IF(COE(I))9,7,9	477
7	N4=N4+1	478

ROOTR(N4)=0.	479
ROOTI(N4)=0.	480
I=-1	481
IF(N4-N1)19,37,19	482
9 CONTINUE	483
10 AXR=0.8	484
AXI=0.	485
L=1	486
N3=1	487
ALP1R=AXR	488
ALP1I=AXI	489
M=1	490
GOTO99	491
11 BET1R=TEMR	492
BET1I=TEMI	493
AXR=0.85	494
ALP2R=AXR	495
ALP2I=AXI	496
M=2	497
GOTO99	498
12 RET2R=TEMR	499
RET2I=TEMI	500
AXR=C.9	501
ALP3R=AXR	502
ALP3I=AXI	503
M=3	504
GOTO99	505
13 RET3R=TEMR	506
BET3I=TEMI	507
14 TE1=ALP1R-ALP3R	508
TE2=ALP1I-ALP3I	509
TE5=ALP3R-ALP2R	510
TE6=ALP3I-ALP2I	511
TEM=TE5*TE5+TE6*TE6	512
TE3=(TE1*TE5+TE2*TE6)/TEM	513
TE4=(TE2*TE5-TE1*TE6)/TEM	514
TE7=TE3+1.	515
TE9=TE3*TE3-TE4*TE4	516
TE10=2.*TE3*TE4	517
DE15=TE7*BET3R-TE4*BET3I	518
DE16=TE7*BET3I+TE4*BET3R	519
TE11=TE3*BET2R-TE4*BET2I+BET1R-DE15	520
TE12=TE3*BET2I+TE4*BET2R+BET1I-DE16	521
TE7=TE9-1.	522
TE1=TE9*BET2R-TE10*BET2I	523
TE2=TE9*BET2I+TE10*BET2R	524
TE13=TE1-BET1R-TE7*BET3R+TE10*BET3I	525
TE14=TE2-BET1I-TE7*BET3I-TE10*BET3R	526
TE15=DE15*TE3-DE16*TE4	527
TE16=DE15*TE4+DE16*TE3	528
TE1=TE13*TE13-TE14*TE14-4.*(TE11*TE15-TE12*TE16)	529
TE2=2.*TE13*TE14-4.*(TE12*TE15+TE11*TE16)	530
TEM = SQRT (TE1*TE1+TE2*TE2)	531
IF(TE1)113,113,112	532
113 TE4 = SQRT (.5 * (TEM - TE1))	533
TE3=.5*TE2/TE4	534
GO TO 111	535
112 TE3 = SQRT (.5 * (TEM + TE1))	536
IF(TE2)110,200,200	537
110 TE3=-TE3	538

200	TE4=.5*TE2/TE3	539
111	TE7=TE13+TE3	540
	TE8=TE14+TE4	541
	TE9=TE13-TE3	542
	TE10=TE14-TE4	543
	TE1=2.*TE15	544
	TE2=2.*TE16	545
	IF(TE7*TE7+TE8*TE8-TE9*TE9-TE10*TE10)204,204,205	546
204	TE7=TE9	547
	TE8=TE10	548
205	TEM=TE7*TE7+TE8*TE8	549
	TE3=(TE1*TE7+TE2*TE8)/TEM	550
	TE4=(TE2*TE7-TE1*TE8)/TEM	551
	AXR=ALP3R+TE3*TE5-TE4*TE6	552
	AXI=ALP3I+TE3*TE6+TE4*TE5	553
	ALP4R=AXR	554
	ALP4I=AXI	555
	M=4	556
	GO TO 99	557
15	N6=1	558
38	IF (ABS (HELL) + ABS (BELL) - 1.E-20) 18,18,16	559
16	TE7 = ABS (ALP3R - AXR) + ABS (ALP3I - AXI)	560
	IF (TE7 / (ABS (AXR) + ABS (AXI)) - 1.E-7)18,18,17	561
17	N3=N3+1	562
	ALP1R=ALP2R	563
	ALP1I=ALP2I	564
	ALP2R=ALP3R	565
	ALP2I=ALP3I	566
	ALP3R=ALP4R	567
	ALP3I=ALP4I	568
	BET1R=BET2R	569
	BET1I=BET2I	570
	BET2R=BET3R	571
	BET2I=BET3I	572
	BET3R=TEMR	573
	BET3I=TEMI	574
	IF(N3-100)14,18,18	575
18	N4=N4+1	576
	ROOTR(N4)=ALP4R	577
	ROOTI(N4)=ALP4I	578
	N3=0	579
41	IF(N4-N1)30,37,37	580
37	RETURN	581
30	IF (ABS (ROOTI(N4)) - 1.E-5)10,10,31	582
31	GO TO(32,10),L	583
32	AXR=ALP1R	584
	AXI=-ALP1I	585
	ALP1I=-ALP1I	586
	M=5	587
	GO TO 99	588
33	BET1R=TEMR	589
	BET1I=TEMI	590
	AXR=ALP2R	591
	AXI=-ALP2I	592
	ALP2I=-ALP2I	593
	M=6	594
	GO TO 99	595
34	BET2R=TEMR	596
	BET2I=TEMI	597
	AXR=ALP3R	598

AXI=-ALP3I	599
ALP3I=-ALP3I	600
L=2	601
M=3	602
99 TEMR=COE(1)	603
TEM1=0.0	604
DO100I=1,N1	605
TE1=TEMR*AXR-TEM1*AXI	606
TEM1=TEM1*AXR+TEMR*AXI	607
100 TEMR=TE1+COE(1+1)	608
HELL=TEMR	609
HELL=TEM1	610
42 IF(N4)102,103,102	611
102 DO101I=1,N4	612
TEM1=AXR-ROOT1(I)	613
TEM2=AXI-ROOT1(I)	614
TE1=TEM1*TEM1+TEM2*TEM2	615
TE2=(TEMR*TEM1+TEM1*TEM2)/TE1	616
TEM1=(TEM1*TEM1-TEMR*TEM2)/TE1	617
101 TEMR=TE2	618
103 GO TO(11,12,13,15,33,34),M	619
END	620
C	621
C	622
C	623
C	624
SUBROUTINE EXPLOT(XARRAY,ICIMEN,YARRAY,KDIMEN,BLIMIT,LET)	625
INTEGER POINT(130,100),GRAPHL(120,58),STAR	626
DIMENSION XARRAY(KDIMEN),YARRAY(KDIMEN)	627
DIMENSION ENCRM1(120),ENCRM2(58)	628
DIMENSION SAVE1(100,100),SAVE2(100,100)	629
COMMON SAVE1,SAVE2,POINT	630
EQUIVALENCE(POINT(1,1),GRAPHL(1,1))	631
DATA STAR/1H+ /	632
WRITE(6,1)	633
READ(5,10)D1,D2	634
D3=BLIMIT	635
D4=-BLIMIT	636
IF(D2.GT.D1) CALL SWITCH(C1,D2)	637
IF(D4.GT.D3) CALL SWITCH(C3,D4)	638
IC1=D3	639
JDIMEN=KDIMEN	640
CALL BLANKR(GRAPHL,120,58,120)	641
DO 22 I=1,JDIMEN	642
X=XARRAY(I)	643
Y=YARRAY(I)	644
IF(X.EQ.0.0.AND.Y.EQ.0.0) GO TO 22	645
CALL LINAR(X,LIMITX,120,C1,D2,ENCRM1)	646
CALL LINAR(Y,LIMITY,58,D3,D4,ENCRM2)	647
IF(LIMITX.EQ.0.0.OR.LIMITY.EQ.0) GO TO 22	648
LIMITX=121-LIMITX	649
GRAPHL(LIMITX,LIMITY)=STAR	650
22 CONTINUE	651
WRITE(6,30) IC1,LET,(GRAPHL(I,1),I=1,120)	652
IC1=D4	653
WRITE(6,34)((GRAPHL(I,J),I=1,120),J=2,57)	654
WRITE(6,30) IC1,LET,(GRAPHL(I,58),I=1,120)	655
WRITE(6,35) D2,D1	656
1 FORMAT(1H1)	657
10 FORMAT(6F10,0,20X)	658

30	FORMAT(1X,I4,A4,1X,120A1)	659
34	FORMAT(10X,120A1)	660
35	FORMAT(3X,1P1E10.3,7HRAD/SEC,91X,1P1E10.3,7HRAD/SEC)	661
61	RETURN	662
	END	663
C		664
C		665
C		666
C		667
	SUBROUTINE LINAR (Y,LIMITY,IDIMEN,DIMNS1,DIMNS2,ENCRMT)	668
	DIMENSION ENCRMT(IDIMEN)	669
	DIMENS=DIMNS1-DIMNS2	670
	LIMITY=0	671
	J=IDIMEN-1	672
	A=J	673
	DELTA=DIMENS/A	674
	ENCRMT(1)=DIMNS1	675
	ENCRMT(IDIMEN)=DIMNS2	676
	DO 10 I=2,J	677
10	ENCRMT(I)=ENCRMT(I-1)-DELTA	678
	DO 11 I=1,J	679
11	IF(Y.LE.ENCRMT(I).AND.Y.GE.ENCRMT(I+1))GO TO 20	680
	GO TO 40	681
20	LIMITY=1	682
40	RETURN	683
	END	684
C		685
C		686
	SUBROUTINE SWITCH(X,Y)	687
	TEST=Y	688
	Y=X	689
	X=TEST	690
	RETURN	691
	END	692
C		693
C		694
	SUBROUTINE ERCHEK(X,I)	695
	DIMENSION X(100)	696
	DATA ERR LIM/0.1E-8/	697
	DO 10 J=1,I	698
10	IF(ABS(X(J)).LT.ERR LIM) X(J)=0.0	699
	RETURN	700
	END	701
C		702
C		703
C		704
C		705
	SUBROUTINE EXPAND(ILOVE,X,Y)	706
	DIMENSION SAVE1(100,100),SAVE2(100,100),X2(2000),Y2(2000),X(2000),	707
	* Y(2000),X1(2000),Y1(2000),Z(125),Z1(87)	708
	INTEGER SPOT(130,50),DASH,C,BLANK,PCINT(130,100)	709
	DATA C/1H1/, DASH/1H-/,BLANK/1H /	710
	DATA APLUS/0.0/,BPLUS/0.0/,AMINUS/0.0/,BMINUS/0.0/	711
	COMMON /INFO4/ C,BLANK,DASH	712
	COMMON /PAYNE/AM,AP,BM,BP	713
	COMMON /PJ/X1,Y1	714
	COMMON SAVE1,SAVE2,POINT	715
	EQUIVALENCE(AP,APLUS),(AM,AMINUS),(BM,BMINUS),(BP,BPLUS)	716
	EQUIVALENCE(X1(1),Z(1)),(Y1(1),Z1(1))	717
	EQUIVALENCE (SPOT(1,1),POINT(1,1)),(11,12)	718

DO 40 J=1,ILOVE	719
DO 50 M=1,2000	720
X1(M)=0.0	721
50 Y1(M)=0.0	722
CALL BLANKR(SPOT,130,50,126)	723
DO 54 M=1,2000	724
X2(M)=0.0	725
54 Y2(M)=0.0	726
READ(5,10) OMEGA,ENCRMT,SIGMA,DELTA	727
OMEGA=ABS(OMEGA)	728
10 FORMAT(4F10.0)	729
A=ENCRMT	730
PERCNT=A*0.01	731
B=OMEGA	732
B=B*PERCNT	733
APLUS=B+OMEGA	734
D=AMINUS	737
I1=1	738
AMINUS=OMEGA-B	735
C=APLUS	736
DO 41 L=1,2000	739
IF(Y(L).LT.D)GO TO 41	740
IF(Y(L).GT.C)GO TO 41	741
23 X1(I1)=X(L)	742
Y1(I1)=Y(L)	743
I1=I1+1	744
41 CONTINUE	745
A=DELTA	746
901 PERCNT=A*0.01	747
B=SIGMA	748
903 B=B*PERCNT	749
BPLUS=SIGMA+B	750
BMINUS=SIGMA-B	751
C=BPLUS	752
D=BMINUS	753
I2=1	754
910 DO 42 L=1,2000	755
IF(C.GT.0.0) GO TO 60	756
IF(X1(L).GT.D) GO TO 42	757
IF(X1(L).LT.C) GO TO 42	758
GO TO 25	759
60 IF(X1(L).GT.C) GO TO 42	760
IF(X1(L).LT.D) GO TO 42	761
25 X2(I2)=X1(L)	762
Y2(I2)=Y1(L)	763
I2=I2+1	764
42 CONTINUE	765
CALL SPLIT(X2,Y2,SPOT,APLUS,AMINUS,BPLUS,BMINUS)	766
40 CONTINUE	767
RETURN	768
END	769
C	770
C	771
C	772
C	773
SUBROUTINE SPLIT(X,Y,SPOT,APLUS,AMINUS,BPLUS,BMINUS)	774
DIMENSION X1(2000),Y1(2000)	775
DIMENSION X(2000),Y(2000),Z(125),Z1(87)	776
INTEGER SPOT(130,50)	777
COMMON /PJ/X1,Y1	778

EQUIVALENCE(X1(1),Z(1)),(Y1(1),Z1(1))	779
A=APLUS	780
B=AMINUS	781
C=A-B	782
D=BPLUS	783
E=BMINUS	784
G=ABS(D)	785
H=ABS(E)	786
F=G-H	787
DELTA=C/124.0	788
DIFF=F/50.0	789
DO 11 J=1,124	790
Z(J)=A	791
11 A=A-DELTA	792
Z(125)=B	793
IF(D.LT.0.0)DIFF=-DIFF	794
DO 12 J=1,49	795
Z1(J)=D	796
12 D=D-DIFF	797
Z1(50)=E	798
CALL BRAKUP(Z,X,Y,SPOT,Z1)	799
RETURN	800
END	801
C	802
C	803
C	804
C	805
SUBROUTINE BRAKUP(YY,X,Y,SPCT,XX)	806
LOGICAL SKIP,SKIP1	807
INTEGER SPOT(130,50),	808
XPT,YPT,STAR	809
DATA STAR/1+X/	810
DIMENSION X(2000),Y(2000),YY(125),XX(87)	811
COMMON /PAYNE/AM,AP,BM,BP	812
EQUIVALENCE(AP,APLUS),(AM,AMINUS),(BM,BMINUS),(BP,BPLUS)	813
SKIP=.FALSE.	814
SKIP1=.FALSE.	815
DO 43 J=1,2000	816
DO 43 I=1,125	817
IF(Y(IJ).NE.0.0) GO TO 30	818
IF(X(IJ).NE.0.0) GO TO 30	819
L=J+1	820
M=J+6	821
DO 60 N=L,M	822
IF(N.GT.2000) GO TO 60	823
IF(Y(N).NE.0.0) GO TO 30	824
IF(X(N).NE.0.0) GO TO 30	825
60 CONTINUE	826
GO TO 40	827
30 IF(SKIP) GO TO 20	828
IF(Y(J).LT.YY(1)) GO TO 20	829
YPT=I	830
SKIP=.TRUE.	831
20 IF(SKIP1)GO TO 48	832
IF(I.GT.50) GO TO 43	833
IF(BPLUS.GT.0.0) GO TO 10	834
IF(X(J).GT.XX(1)) GO TO 43	835
GO TO 11	836
10 IF(X(J).LT.XX(1)) GO TO 43	837
11 IF(BPLUS.LT.0.0) XPT=I	838
IF(BPLUS.GE.0.) XPT=51-I	

SKIPI=.TRUE.	839
IF(.NOT.SKIPI)GO TO 43	840
50 I=125	841
SPOT(YPT,XPT)=STAR	842
SKIP=.FALSE.	843
SKIPI=.FALSE.	844
48 IF(SKIPI)GO TO 50	845
43 CONTINUE	846
40 CALL RITEIT(SPOT)	847
RETURN	848
END	849
C	850
C	851
C	852
C	853
SUBROUTINE RITEIT(SPOT)	854
COMMON /PAYNE/AM,AP,BM,BP	855
EQUIVALENCE(AP,APLUS),(AM,AMINUS),(BM,BMINUS),(BP,BPLUS)	856
INTEGER SPOT(130,50)	857
IF(BP.LT.BM) GO TO 40	858
A=BP	859
B=BM	860
WRITE(6,1)	861
1 FORMAT(1H1,40X,43HCOMPLEX FREQUENCY PLANE,RIGHT HAND QUADRANT)	862
GO TO 30	863
40 A=BM	864
B=BP	865
WRITE(6,2)	866
2 FORMAT(1H1,40X,43HCOMPLEX FREQUENCY PLANE, LEFT HAND QUADRANT)	867
30 WRITE(6,12)AP,AM	868
12 FORMAT(114X,15HC-----J-OMEGA,/1X,F8.2,112X,F8.2)	869
WRITE(6,14)B	870
14 FORMAT(124X,5HSIGMA,/126X,1HI,/126X,1HV,/121X,F8.2)	871
WRITE(6,11) SPOT	872
11 FORMAT(1X,130A1)	873
WRITE(6,15)A	874
15 FORMAT(60X,27HLINEAR EXPANC PLOT(RAD/SEC) ,34X,F8.2)	875
RETURN	876
END	877
C	878
C	879
SUBROUTINE SAVER(ROOTR,ROOTI,IDIC,SAVE1,SAVE2,JZO,K1)	880
DIMENSION SAVE1(100,100),SAVE2(130,100),ROOTR(100),ROOTI(100)	881
IF(K1)30,9,10	882
9 K1=1	883
10 IDI=IDIC	884
IDIC=IDIC+(K1-1)	885
IF(IDIC.GE.100)GO TO 30	886
50 DO 40 IZAP=K1,IDIC	887
IZA=IZAP-(K1-1)	888
SAVE1(JZO,IZAP)=ROOTR(IZA)	889
SAVE2(JZO,IZAP)=ROOTI(IZA)	890
40 CONTINUE	891
K1=IDIC+1	892
GO TO 20	893
30 JZO=JZO+1	894
IDIC=IDI	895
GO TO 9	896
20 RETURN	897
END	898

C
C
C
C

```
      SUBROUTINE PLOTTER(SAVE1,SAVE2,ANUMB1,ANUMB2,POINT,XA,XB,PI,PRNT,  
      *STAR,NBB,NO)  
      DIMENSION SAVE1(100,100),SAVE2(100,100),XA(2000),XB(2000)  
      INTEGER POINT(130,100),DASH,BLANK,STAR,ROTLCS,EXPND,FREQSR,DENSE  
      COMMON /INFO7/ ICOUNT  
      COMMON /INFO8/ROTLCS,EXPND,FREQSR  
      COMMON /INFO9/ DENSE  
      LOGICAL PRNT  
      IF(PRNT) GO TO 43  
      DO 40 NAR=1,100  
      DO 40 NBR=1,100  
      IF(SAVE1(NAB,NBR))41,42,41  
42 IF(SAVE2(NAB,NBR))41,40,41  
41 ANUMB1=SAVE1(NAB,NBR)  
      ANUMB2=SAVE2(NAB,NBR)  
      ANUMB2=ABS(ANUMB2)  
43 AZZ=ANUMB2  
      AZZ=AZZ*100000.0  
      NZZ=AZZ  
      IF(NZZ.EQ.0) GO TO 45  
50 NBB=NBB+1  
45 CALL EXCUTE(ANUMB1,ANUMB2,POINT,XA,XB,MI,NO,STAR)  
      IF (DENSE.NE.1) GO TO 10  
      ICOUNT=ICOUNT+1  
      GO TO 11  
10 IF(PRNT) ICOUNT=ICOUNT+1  
11 IF(PRNT) RETURN  
40 CONTINUE  
      IF(ROTLCS.EQ.0) RETURN  
70 CALL WRITIT(XA,XB)  
      CALL PREPAR(POINT)  
      RETURN  
      END
```

C
C
C
C

```
      SUBROUTINE EXCUTE(ANUMB1,ANUMB2,POINT,XA,XB,MI,NO,STAR)  
      LOGICAL SKIP1,SKIP2,LESS  
      INTEGER ROTLCS,EXPND,FREQSR  
      DIMENSION XA(2000),XB(2000)  
      INTEGER POINT(130,100),DASH,BLANK,STAR  
      COMMON /INFO8/ROTLCS,EXPND,FREQSR  
      DATA      K1/10/,K2/100/,K3/1000/,K4/10000/,NEGONE/-1/  
      LESS=.FALSE.  
      I=0  
      J=0  
      XA(NO)=ANUMB1  
      XB(NO)=ABS(ANUMB2)  
      NO=NO+1  
      IF(ROTLCS.EQ.0) RETURN  
      IF(ABS(ANUMB1).GT.10000.0) GO TO 50  
      IF(ABS(ANUMB2).GT.10000.0) GO TO 50  
      IF(ABS(ANUMB1).EQ.0.0) GO TO 500  
      IF(ABS(ANUMB2).EQ.0.0) GO TO 500  
      IF(ABS(ANUMB2).LT.0.001) GO TO 50
```

	IF(ABS(ANUMB1).LT.0.001) GO TO 50	959
500	CALL SCALE1(K1,K2,K3,K4,NEGONE,I,ANUMB1,ICONS,LESS,SKIP1)	960
	CALL SCALE2(K1,K2,K3,K4,J,JCONS,ANUMB2,SKIP2)	961
	CALL WPOINT(J,JCONS,L,ANUMB2,SKIP2)	962
	CALL SPOINT(I,ANUMB1,LESS,ICONS,L,POINT,SKIP1,NO,XA,XB,STAR)	963
50	RETURN	964
	END	965
C		966
C		967
C		968
C		969
	SUBROUTINE SCALE1(K1,K2,K3,K4,NEGONE,I,ANUMB1,ICONS,LESS,SKIP1)	970
	LOGICAL LESS	971
	LOGICAL SKIP1	972
	SKIP1=.FALSE.	973
	ICONS=1	974
	I=1	975
	AKEEP=ANUMB1	976
21	NUMB1=ANUMB1	977
	NUMB=IABS(NUMB1)	978
	IF(NUMB.EQ.0)GOTO 12	979
	IF(NUMB1)31,40,40	980
40	IF(NUMB1.GE.10)GOTO 11	981
	GO TO 50	982
31	LESS=.TRUE.	983
	XXX=-NUMB1	984
	NUMB1=XXX	985
	GOTO 40	986
12	GOTO(1,2,3,4),I	987
1	ANUMB1=AKEEP	988
	ICONS=K1	989
	RK1=K1	990
	ANUMB1=ANUMB1*RK1	991
	I=2	992
	GOTO 21	993
2	ANUMB1=AKEEP	994
	ICONS=K2	995
	RK2=K2	996
	ANUMB1=ANUMB1*RK2	997
	I=3	998
	GOTO 21	999
3	ANUMB1=AKEEP	1000
	ICONS=K3	1001
	RK3=K3	1002
	ANUMB1=ANUMB1*RK3	1003
	I=4	1004
	GOTO 21	1005
4	ANUMB1=AKEEP	1006
	ICONS=K4	1007
	RK4=K4	1008
	ANUMB1=ANUMB1*RK4	1009
	NUMB1=ANUMB1	1010
	IF(NUMB1.EQ.0) GO TO 51	1011
	GO TO 50	1012
11	SKIP1=.TRUE.	1013
	GO TO (6,7,8,9),I	1014
6	ANUMB1=AKEEP	1015
	ICONS=K1	1016
	RK1=K1	1017
	ANUMB1=ANUMB1/RK1	1018

I=2	1019
GOTO 21	1020
7 ANUMB1=AKEEP	1021
I CONS=K2	1022
RK2=K2	1023
ANUMB1=ANUMB1/RK2	1024
I=3	1025
GOTO 21	1026
8 ANUMB1=AKEEP	1027
I CONS=K3	1028
RK3=K3	1029
ANUMB1=ANUMB1/RK3	1030
I=4	1031
GOTO 21	1032
9 ANUMB1=AKEEP	1033
I CONS=K4	1034
RK4=K4	1035
ANUMB1=ANUMB1/RK4	1036
GO TO 50	1037
51 I=5	1038
50 RETURN	1039
END	1040
	1041
	1042
	1043
	1044
SUBROUTINE SCALE2(K1,K2,K3,K4,J,JCONS,ANUMB2,SKIP2)	1045
LOGICAL SKIP2	1046
SKIP2=.FALSE.	1047
JCONS=1	1048
J=1	1049
ANUMB2=ABS(ANUMB2)	1050
BKEEP=ANUMB2	1051
20 NUMB2=ANUMB2	1052
IF(NUMB2.EQ.0)GO TO 10	1053
IF(NUMB2.GE.10)GO TO 11	1054
GO TO 50	1055
10 GOTO(1,2,3,4),J	1056
1 ANUMB2=BKEEP	1057
JCONS=K1	1058
RK1=K1	1059
ANUMB2=ANUMB2*RK1	1060
J=2	1061
GO TO 20	1062
2 ANUMB2=BKEEP	1063
JCONS=K2	1064
RK2=K2	1065
ANUMB2=ANUMB2*RK2	1066
J=3	1067
GO TO 20	1068
3 ANUMB2=BKEEP	1069
JCONS=K3	1070
RK3=K3	1071
ANUMB2=ANUMB2*RK3	1072
J=4	1073
GO TO 20	1074
4 ANUMB2=BKEEP	1075
JCONS=K4	1076
RK4=K4	1077
ANUMB2=ANUMB2*RK4	1078

NUMB2=ANUMB2	1079
IF(NUMB2.EQ.0) GO TO 51	1080
GO TO 50	1081
11 SKIP2=.TRUE.	1082
GO TO (6,7,8,9),J	1083
6 JCONS=K1	1084
ANUMB2=BKEEP	1085
RK1=K1	1086
ANUMB2=ANUMB2/RK1	1087
J=2	1088
GO TO 20	1089
7 JCONS=K2	1090
ANUMB2=BKEEP	1091
RK2=K2	1092
ANUMB2=ANUMB2/RK2	1093
J=3	1094
GO TO 20	1095
8 JCONS=K3	1096
ANUMB2=BKEEP	1097
RK3=K3	1098
ANUMB2=ANUMB2/RK3	1099
J=4	1100
GO TO 20	1101
9 JCONS=K4	1102
ANUMB2=BKEEP	1103
RK4=K4	1104
ANUMB2=ANUMB2/RK4	1105
GO TO 50	1106
51 J=5	1107
50 RETURN	1108
END	1109
C	1110
C	1111
C	1112
C	1113
SUBROUTINE WPOINT(J,JCONS,L,ANUMB2,SKIP2)	1114
LOGICAL SKIP2	1115
L=0	1116
IDELTA=0	1117
IF(ANUMB2.GE.9.2) IDELTA=12	1118
IF(ANUMB2.GE.8.0) GO TO 200	1119
IF(ANUMB2.GE.6.9) GO TO 201	1120
IF(ANUMB2.GE.5.9) GO TO 202	1121
IF(ANUMB2.GE.5.0) GO TO 203	1122
IF(ANUMB2.GE.4.2) GO TO 204	1123
IF(ANUMB2.GE.3.5) GO TO 205	1124
IF(ANUMB2.GE.2.9) GO TO 206	1125
IF(ANUMB2.GE.2.4) GO TO 207	1126
IF(ANUMB2.GE.1.9) GO TO 208	1127
IF(ANUMB2.GE.1.5) GO TO 209	1128
IF(ANUMB2.GE.1.2) GO TO 210	1129
GO TO 211	1130
200 IF(ANUMB2.LT.9.2) IDELTA=11	1131
201 IF(ANUMB2.LT.8.0) IDELTA=10	1132
202 IF(ANUMB2.LT.6.9) IDELTA=9	1133
203 IF(ANUMB2.LT.5.9) IDELTA=8	1134
204 IF(ANUMB2.LT.5.0) IDELTA=7	1135
205 IF(ANUMB2.LT.4.2) IDELTA=6	1136
206 IF(ANUMB2.LT.3.5) IDELTA=5	1137
207 IF(ANUMB2.LT.2.9) IDELTA=4	1138

208	IF(ANUMB2.LT.2.4) IDELTA=3	1139
209	IF(ANUMB2.LT.1.9) IDELTA=2	1140
210	IF(ANUMB2.LT.1.5) IDELTA=1	1141
211	IF(ANUMB2.LT.1.2) IDELTA=0	1142
	IDELTA=13-IDELTA	1143
	IF(J.EQ.5) GO TO 50	1144
	IF(SKIP2) GO TO 41	1145
	IF(JCONS.EQ.1)GOTO 1	1146
	IF(JCONS.EQ.10)GOTO 10	1147
	IF(JCONS.EQ.100)GOTO 100	1148
	IF(JCONS.EQ.1000)GOTO 1000	1149
	IF(JCONS.EQ.10000)GOTO 10000	1150
1	L=73+IDELTA	1151
	GO TO 40	1152
10	L=86+IDELTA	1153
	GO TO 40	1154
100	L=99+IDELTA	1155
	GO TO 40	1156
1000	L=112+IDELTA	1157
	GO TO 40	1158
10000	L=125	1159
	GO TO 40	1160
41	KCONS=JCONS+1	1161
	SKIP2=.FALSE.	1162
	IF(KCONS.EQ.11)GOTO 11	1163
	IF(KCONS.EQ.101)GOTO 101	1164
	IF(KCONS.EQ.1001)GOTO 1001	1165
	IF(KCONS.EQ.10001)GOTO 10001	1166
11	L=60+IDELTA	1167
	GO TO 40	1168
101	L=47+IDELTA	1169
	GO TO 40	1170
1001	L=34+IDELTA	1171
	GO TO 40	1172
50	L=125	1173
	GO TO 40	1174
10001	L=34	1175
40	RETURN	1176
	END	1177
C		1178
C		1179
C		1180
C		1181
	SUBROUTINE SPOINT(I,ANUMB1,LESS,ICONS,L,PCINT,SKIP1,NO,XA,XB,STAR)	1182
	DIMENSION XA(2000),XB(2000)	1183
	INTEGER POINT(130,100),DASH,BLANK,STAR,Q	1184
	INTEGER STAR1,STAR2	1185
	DATA STAR1/1H0/,STAR2/1H+/,	1186
	DATA BLANK/1H /,DASH/1H-/,	1187
	LOGICAL SKIP1,LESS	1188
	AKEEP=ANUMB1	1189
	ANUMB1=ARS(AKEEP)	1190
	KOOL=0	1191
	IF(1.EQ.5)GO TO 1000	1192
	IF(ANUMB1.GE.7.1)KOOL=7	1193
	IF(ANUMB1.GE.5.0)GO TO 1011	1194
	IF(ANUMB1.GE.4.0)GO TO 1111	1195
	IF(ANUMB1.GE.3.1) GO TO 1211	1196
	IF(ANUMB1.GE.2.3)GOTO1311	1197
	IF(ANUMB1.GE.1.6) GO TO 1411	1198

IF(ANUMB1.GE.1.0)GO TO 1511	1199
GO TO 4011	1200
1011 IF(ANUMB1.LT.7.1)K00L=6	1201
1111 IF(ANUMB1.LT.5.0)K00L=5	1202
1211 IF(ANUMB1.LT.4.0)K00L=4	1203
1311 IF(ANUMB1.LT.3.1)K00L=3	1204
1411 IF(ANUMB1.LT.2.3)K00L=2	1205
1511 IF(ANUMB1.LT.1.6)K00L=1	1206
4011 NCRMNT=8-KOGL	1207
IF(LESS)GO TO 40	1208
IF(SKIP1)GO TO 41	1209
IF(ICON5.EQ.1)GOTO 1	1210
IF(ICON5.EQ.10)GOTO 10	1211
IF(ICON5.EQ.100)GOTO 100	1212
IF(ICON5.EQ.1000)GOTO 1000	1213
1 K00L=71+K00L	1214
GO TO 50	1215
10 K00L=64+K00L	1216
GO TO 50	1217
100 K00L=57+K00L	1218
GO TO 50	1219
1000 K00L=50+K00L	1220
GO TO 50	1221
41 LCONS=ICON5+1	1222
SKIP1=.FALSE.	1223
IF(LCONS.EQ.11)GOTO 11	1224
IF(LCONS.EQ.101)GOTO 101	1225
IF(LCONS.EQ.1001)GOTO 1001	1226
11 K00L=78+K00L	1227
GO TO 50	1228
101 K00L=85+K00L	1229
GO TO 50	1230
1001 K00L=92+K00L	1231
50 LESS=.FALSE.	1232
IF(POINT(L,KOOL).EQ.DASH) GO TO 51	1233
IF(POINT(L,KOOL).EQ.C) GO TO 51	1234
14 IF(POINT(L,KOOL).EQ.STAR1) GO TO 15	1235
IF(POINT(L,KOOL).EQ.STAR2) GO TO 15	1236
IF(STAR.EQ.STAR1) GO TO 51	1237
IF(STAR.EQ.STAR2) GO TO 51	1238
IF(POINT(L,KOOL).NE.BLANK) GO TO 70	1239
51 POINT(L,KOOL)=STAR	1240
ANUMB1=AKEEP	1241
GO TO 70	1242
15 L=L+1	1243
IF(L.EQ.130) GO TO 51	1244
GO TO 14	1245
40 IF(SKIP1) GO TO 42	1246
I2=ICON5+2	1247
IF(I2.EQ.3)GOTO 3	1248
IF(I2.EQ.12)GOTO 12	1249
IF(I2.EQ.102)GOTO 102	1250
IF(I2.EQ.1002)GOTO 1002	1251
3 K00L=21+NCRMNT	1252
GO TO 50	1253
12 K00L=28+NCRMNT	1254
GO TO 50	1255
102 K00L=35+NCRMNT	1256
GO TO 50	1257
1002 K00L=42+NCRMNT	1258

GO TO 50	1259
42 IY=I CONS+3	1260
SKIPI=.FALSE.	1261
IF(IY.EQ.13)GOTO 13	1262
IF(IY.EQ.103)GOTO 103	1263
IF(IY.EQ.1003)GOTO 1003	1264
IF(IY.EQ.10003)GOTO 10003	1265
13 KOOL=14+NCRMNT	1266
GO TO 50	1267
103 KOOL=7+NCRMNT	1268
GO TO 50	1269
1003 KOOL=NCRMNT	1270
GO TO 50	1271
10003 POINT(L,1)=STAR	1272
GO TO 70	1273
70 RETURN	1274
END	1275
C	1276
C	1277
C	1278
C	1279
SUBROUTINE PREPAR(POINT)	1280
INTEGER POINT(130,100),PRT	1281
DATA PRT /6/	1282
WRITE(6,1)	1283
1 FORMAT(1H1)	1284
WRITE(6,11)	1285
11 FORMAT(///)	1286
WRITE(PRT,14)	1287
14 FORMAT(75X,8HLOG PLOT,/58X,42HCOMPLEX FREQUENCY PLANE,LEFT HAND CU	1288
*ADRANT,/75X,9H(RAD/SEC))	1289
WRITE(6,12)	1290
12 FORMAT(32X,5H10000,9X,4H1000,9X,3H100,10X,2H10,12X,1H1,11X,2H.1,10	1291
*X,14H.01<---J-OMEGA,/ 18X,11HMINUS SIGMA)	1292
13 FORMAT(32X,5H10000,9X,4H1000,9X,3H100,10X,2H10,12X,1H1,11X,2H.1,10	1293
*X,14H.01<---J-OMEGA,/ 18X,11H PLUS SIGMA)	1294
DO 50 I=1,130	1295
50 POINT(I,50)=POINT(I,51)	1296
WRITE(PRT,10)POINT	1297
10 FORMAT(1X,130A1)	1298
WRITE(6,13)	1299
WRITE(PRT,15)	1300
15 FORMAT(75X,8HLOG PLOT,/58X,43HCOMPLEX FREQUENCY PLANE,RIGHT HAND Q	1301
*UADRANT,/75X,9H(RAD/SEC))	1302
RETURN	1303
END	1304
C	1305
C	1306
C	1306
C	1307
C	1308
SUBROUTINE WRITIT(XA,XB)	1309
DIMENSION XA(2000),XB(2000)	1310
WRITE(6,1)	1311
1 FORMAT(1H1)	1312
WRITE(6,11)	1313
11 FORMAT(5X,48HTHE FOLLOWING ROOTS ARE PLOTTED ON THE LOG PLOT,/	1314
15X,99HROOTS AT THE ORIGIN ARE NOT PRINTED OR PLOTTED, ROOTS ON THE	1315
* J-OMEGA AXIS ARE NOT PLOTTED.	1316
2, //16X,5HSIGMA,25X,7HJ-OMEGA, //)	1317

DO 77 IZ0=1,2000	1318
XC=XB(IZ0)	1319
IF(XC.GT.0.0) XC=-XC	1320
IF(XC.NE.0.0) GO TO 22	1321
20 IF(XA(IZ0))22,40,22	1322
40 K1=IZ0	1323
K2=IZ0+12	1324
DO 50 K=K1,K2	1325
IF(XA(K).NE.0.0) GO TO 77	1326
50 IF(XB(K).NE.0.0) GO TO 77	1327
GO TO 60	1328
22 WRITE(6,10)XA(IZ0),XC	1329
10 FORMAT(5X,F20.9,10X,F20.9)	1330
77 CONTINUE	1331
60 RETURN	1332
END	1333
C	1334
C	1335
SUBROUTINE FREQRS(SIGMAX,CMEGAY)	1336
DIMENSION SAVE1(100,100),SAVE2(100,100),ACOEFF(100),BCOEFF(100),	1337
* AAAAAA(99),BBBBBB(99),SIGMAX(2000),OMEGAY(2000),DEC(9),	1338
* OMEGAS(55),OMEGA(17),WLOG(3000),PHILIN(3000),MDBLIN(3000),	1339
* PHIPLN(3000),MPDBLN(3000),ALIM(7),BIGSEM(3000),	1340
* EMSUBS(3000)	1341
REAL MDBLIN,MPDBLN	1342
INTEGER FRQEXP,IEXP(4)	1343
INTEGER POINT(130,100)	1344
COMMON/FRFEK/ACOEFF,BCOEFF	1345
COMMON/INFG2/ALIM	1346
COMMON/INF03/OMEGA	1347
COMMON/INF06/EMSUBS	1348
COMMON /INF07/ ICOUNT	1349
COMMON SAVE1,SAVE2,POINT	1350
EQUIVALENCE (WLOG(1),SAVE1(1)),(PHILIN(1),SAVE2(1)),	1351
1 (SAVE1(3001),MDBLIN(1)),(SAVE2(3001),PHIPLN(1)),	1352
2 (SAVE1(6001),MPDBLN(1)),(SAVE2(6001),BIGSEM(1)),	1353
3 (ALIM(1),XSMALL),(AAAAAA(1),ACOEFF(2)),(BBBBBB(1),	1354
4 BCOEFF(2))	1355
TENS(X)=X*10.0	1356
DATA(DEC(1),I=1,9)/1.0,2.0,3.0,4.0,5.0,6.0,7.0,8.0,9.0/	1357
DATA LET1/4H CB/,LET2/4H CEG/	1358
K1=0	1359
KOUNT=1	1360
DO 61 I=1,4	1361
61 IEXP(I)=0	1362
DO 60 I=1,3000	1363
62 WLOG(I)=0.0	1364
PHILIN(I)=0.0	1365
BIGSEM(I)=0.0	1366
EMSUBS(I)=0.0	1367
MDBLIN(I)=0.0	1368
PHIPLN(I)=0.0	1369
60 MPDBLN(I)=0.0	1370
WRITE(6,51)	1371
READ(5,19)XSMALL,DEGREE,DBLIM	1372
READ(5,12)FRQEXP	1373
IF(FRQEXP.NE.0)READ(5,12)(IEXP(K),K=1,4)	1374
12 FORMAT(4I10)	1375
19 FORMAT(3E10.0,50X)	1376
DO 90 I=2,7	1377

90 ALIM(I)=TENS(ALIM(I-1))	1378
DO 70 J=1,6	1379
DO 71 I=1,9	1380
K=K1+I	1381
71 OMEGAS(K)=DEC(I)*ALIM(J)	1382
70 K1=K1+9	1383
OMEGAS(55)=ALIM(7)	1384
K=1	1385
CALL FIGURR(55,OMEGAS,KOUNT)	1386
41 Y=ABS(OMEGAY(K))	1387
X=ABS(SIGMAX(K))	1388
IF(Y.EQ.0.0) GO TO 13	1389
YOMEGA=Y	1390
GO TO 20	1391
13 IF(X.EQ.0.0) GO TO 10	1392
YOMEGA=X	1393
20 CALL SWEEPR(YOMEGA)	1394
CALL FIGURR(17,OMEGA,KOUNT)	1395
IF(KOUNT.GE.2984) GO TO 40	1396
GO TO 80	1397
10 KK=K+1	1398
KI=K+11	1399
IF(KI.GT.2000) GO TO 40	1400
DO 11 L=KK,KI	1401
Y=ABS(OMEGAY(L))	1402
X=ABS(SIGMAX(L))	1403
IF(X.NE.0.0.OR.Y.NE.0.0) GO TO 80	1404
11 CONTINUE	1405
GO TO 40	1406
80 IF(K.GE.ICOUNT) GO TO 40	1407
K=K+1	1408
GO TO 41	1409
40 DO 68 I=1,3000	1410
IF(WLOG(I).EQ.0.0.AND.PHILIN(I).EQ.0.0.AND.PHIPLN(I).EQ.C.0.AND.	1411
1MDBLIN(I).EQ.0.0.AND.MPDBLN(I).EQ.C.0.AND.RIGSEM(I).EQ.C.0.AND.	1412
2EMSUBS(I).EQ.0.0) GO TO 69	1413
JJ=MCD(I,57)	1414
IF(JJ.EQ.0) WRITE(6,51)	1415
68 WRITE(6,50) WLOG(I),RIGSEM(I),MDBLIN(I),PHILIN(I),EMSUBS(I),	1416
1MPDBLN(I),PHIPLN(I)	1417
69 WRITE(6,14)	1418
CALL NEWPLT(DEGREE,WLOG,PHILIN,LET2)	1419
IF(IEXP(1).EQ.0) GO TO 30	1420
JJ=IEXP(1)	1421
DO 34 I=1,JJ	1422
34 CALL EXPLOT(WLOG,3000,PHILIN,3000,DEGREE,LET2)	1423
30 CALL NEWPLT(DBLIM,WLOG,MDBLIN,LET1)	1424
IF(IEXP(2).EQ.0) GO TO 31	1425
JJ=IEXP(2)	1426
DO 35 I=1,JJ	1427
35 CALL EXPLOT(WLOG,3000,MDBLIN,3000,DBLIM,LET1)	1428
31 WRITE(6,15)	1429
CALL NEWPLT(DEGREE,WLOG,PHIPLN,LET2)	1430
IF(IEXP(3).EQ.0) GO TO 32	1431
JJ=IEXP(3)	1432
DO 36 I=1,JJ	1433
36 CALL EXPLOT(WLOG,3000,PHIPLN,3000,DEGREE,LET2)	1434
32 CALL NEWPLT(DBLIM,WLOG,MPDBLN,LET1)	1435
IF(IEXP(4).EQ.0) GO TO 33	1436
JJ=IEXP(4)	1437

DO 37 I=1,JJ	1438
37 CALL EXPLOT(WLOG,3000,MPDBLN,3000,DBLIN,LET1)	1439
51 FORMAT(1H1,5X,5HOMEGA,13X,3H/M/,16X,6H/H/-DB,10X,3HPHI,20X,4H/M/P,	1440
*10X,7H/M/P-DB,14X,5HPHI-P,/))	1441
50 FORMAT(2X,1P1E12.5,5X,1P1E12.5,6X,1P1E12.5,5X,1P1E12.5,11X,	1442
*1P1E12.5,3X,1P1E12.5,9X,1P1E12.5)	1443
14 FORMAT(1H1,////////////////////,25X,60HTHE FOLLOWING PLOTS ARE NOR	1444
*MAL FREQUENCY RESPONSE DATA)	1445
15 FORMAT(1H1,////////////////////,25X,36HTHE FOLLOWING PLOTS ARE PCP	1446
*DV DATA)	1447
33 RETURN	1448
END	1449
C	1450
C	1451
SUBROUTINE FIGURR(KONT,OMEGA,KOUNT)	1452
DIMENSION SAVE1(100,100),SAVE2(100,100)	1453
DIMENSION ACOEFF(100),BCOEFF(100),AAAAAA(99),BBBBBB(99)	1454
DIMENSION WLOG(3000),PHILIN(3000),MDBLIN(3000),PHIPLN(3000),	1455
*MPDBLN(3000),BIGSEM(3000),EMSUBS(3000),OMEGA(KONT)	1456
REAL MDBLIN,MPDBLN	1457
INTEGER POINT(130,100)	1458
COMMON/FREK/ACOEFF,BCOEFF	1459
EQUIVALENCE(AAAAAA(1),ACOEFF(2)),(BBBBBB(1),BCOEFF(2))	1460
COMMON/INFO6/EMSUBS	1461
COMMON SAVE1,SAVE2,POINT	1462
EQUIVALENCE (WLOG(1),SAVE1(1)),(PHILIN(1),SAVE2(1)),	1463
1 (SAVE1(3001),MDBLIN(1)),(SAVE2(3001),PHIPLN(1)),	1464
2 (SAVE1(6001),MPDBLN(1)),(SAVE2(6001),BIGSEM(1))	1465
DO 21 I=1,KONT	1466
CALL FIGURE(OMEGA(I),ACOEFF(1),AAAAAA,EMOFAS,PHIAS)	1467
CALL FIGURE(OMEGA(I),PCOEFF(1),BBBBBB,EMOFBS,PHIBS)	1468
BIGEMM=ABS(EMOFAS/EMOFBS)	1469
BIGSEM(KOUNT)=BIGEMM	1470
WLOG(KOUNT)=OMEGA(I)	1471
PHIEND=PHIAS-PHIBS	1472
IF(PHIEND.LT.-180.) PHIEND=PHIEND+360.	1473
IF(PHIEND.GT. 180.) PHIEND=PHIEND-360.	1474
PHILIN(KOUNT)=PHIEND	1475
PHIEND = PHIEND * 0.0174533	1476
REEL = BIGEMM * COS (PHIEND)	1477
EIMAG= BIGEMM * SIN (PHIEND)	1478
EMSUBP = SQRT ((REEL** 2) + ((OMEGA (I) * EIMAG) ** 2))	1479
EMSUBS(KOUNT)=EMSUBP	1480
QUANZ=OMEGA(I) * EIMAG	1481
PHIPEE=ATAN2(QUANZ,REEL)	1482
PHIPEE = PHIPEE * 57.2957795131	1483
IF(PHIPEE . LT . 0.) PHIPEE = PHIPEE + 360.	1484
IF(PHIPEE.LT.-180.) PHIPEE=PHIPEE+360.	1485
IF(PHIPEE.GT. 180.) PHIPEE=PHIPEE-360.	1486
PHIPLN(KOUNT)=PHIPEE	1487
DEEBEE=20.0*(ALOG10(BIGEMM))	1488
MDBLIN(KOUNT)=DEEBEE	1489
DEEBEP=20.0*(ALOG10(EMSUBP))	1490
MPDBLN(KOUNT)=DEEBEP	1491
KOUNT=KOUNT+1	1492
21 CONTINUE	1493
RETURN	1494
END	1495
C	1496
C	1497

SUBROUTINE FIGURE(X,Y,Z,EMOFAS,PHIAS)	1498
DIMENSION Z(99)	1499
REELAS=Y	1500
I=2	1501
A=-1.0	1502
21 REELAS=((Z(I)*(X**I))*A)+REELAS	1503
25 I=I+2	1504
IF(I.GT.99) GO TO 22	1505
A=-A	1506
IF(Z(I).EQ.0.)GO TO 25	1507
GO TO 21	1508
22 I=1	1509
A=1.0	1510
AIMAGA=0.0	1511
24 AIMAGA=(Z(I)*(X**I)*A)+AIMAGA	1512
26 I=I+2	1513
IF(I.GT.99) GO TO 23	1514
A=-A	1515
IF(Z(I).EQ.0.)GO TO 26	1516
GO TO 24	1517
23 EMOFAS=SQRT((REELAS**2)+(AIMAGA**2))	1518
PHIAS=ATAN2(AIMAGA,REELAS)	1519
PHIAS = PHIAS*57.2957795131	1520
IF(PHIAS.LT.0.) PHIAS=PHIAS+360.	1521
RETURN	1522
END	1523
C	1524
C	1525
SUBROUTINE LOGGER(XX,LIMITW)	1526
DIMENSION ALIM(7),DIMS(20)	1527
EQUIVALENCE(WASTE,AXX)	1528
COMMON/INFO2/ALIM	1529
TENS(X)=X*10.0	1530
DATA (DIMS(I),I=1,20) /1.0,1.15,1.3,1.5,1.7,1.95,2.25,2.5,2.85,	1531
*3.15,3.5,3.95,4.5,4.85,5.4,6.05,6.8,7.7,8.8,9.999999/	1532
LIMITW=0	1533
LIMITX=0	1534
LIMIT2=0	1535
DO 20 I=1,6	1536
AA=2*(I-1)	1537
20 IF(XX.LT.ALIM(I+1).AND.XX.GE.ALIM(I)) GO TO 21	1538
RETURN	1539
21 LIMITX=TFNS(AA)	1540
WASTE=ABS(XX)	1541
IF(WASTE.EQ.0.0) RETURN	1542
30 IF(AXX.GE.1.0.AND.AXX.LT.10.0) GO TO 22	1543
IF(AXX.LT.10.0)GO TO 23	1544
AXX=AXX/10.0	1545
GO TO 30	1546
23 AXX=TENS(AXX)	1547
GO TO 30	1548
22 DO 24 I=1,19	1549
24 IF(AXX.GE.DIMS(I).AND.AXX.LT.DIMS(I+1)) GO TO 25	1550
25 LIMIT2=I	1551
LIMITW=LIMITX+LIMIT2	1552
RETURN	1553
END	1554
C	1555
C	1556
SUBROUTINE BLANKR (POINT,N1,N2,LIM1)	1557

INTEGER POINT(N1,N2),BLANK,DASH,Q	1558
COMMON /INFO4/ Q,BLANK,DASH	1559
DATA BLANK/1H /,DASH/1H-/,Q/1H1/	1560
DO 51 M=1,N1	1561
DO 51 N=1,N2	1562
51 POINT(M,N)=BLANK	1563
DO 52 M=1,N1	1564
POINT(M,1)=DASH	1565
52 POINT(M,N2)=DASH	1566
DO 53 M=1,N2	1567
POINT(1,M)=Q	1568
53 POINT(LIM1,M)=Q	1569
RETURN	1570
END	1571
C	1572
C	1573
SUBROUTINE NEWPLT(X,Y,Z,LET)	1574
INTEGER POINT(130,100),POINTF(120,58),STAR,Q	1575
DIMENSION ALIM(7)	1576
DIMENSION Y(3000),Z(3000)	1577
DIMENSION SAVE1(100,100),SAVE2(100,100)	1578
COMMON /INFO4/ Q,BLANK,DASH	1579
COMMON/INFO2/ALIM	1580
COMMON SAVE1,SAVE2,POINT	1581
EQUIVALENCE(POINT(1,1),POINTF(1,1))	1582
DATA STAR/1H*/,Q/1H1/	1583
1 FORMAT(1H1)	1584
CALL BLANKR(POINTF,120,58,120)	1585
DO 10 I=1,3000	1586
IF(Y(I).EQ.0.0) GO TO 11	1587
GO TO 12	1588
11 IF(Z(I).EQ.0.0) GO TO 10	1589
12 CALL LOGGER(Y(I),LIMITW)	1590
IF(LIMITW.EQ.0) GO TO 10	1591
CALL LINEAR(X,LIMITY,Z(I))	1592
IF(LIMITY.EQ.0) GO TO 10	1593
POINTF(LIMITW,LIMITY)=STAR	1594
10 CONTINUE	1595
DO 40 I=20,100,20	1596
DO 40 J=1,58	1597
40 POINTF(I,J)=Q	1598
WRITE(6,1)	1599
30 FORMAT(1X,6(1P1E10.1,1X,3HR/S,6X),1P1E10.1)	1600
J=X	1601
WRITE(6,51)J,LET,(POINTF(1,1),I=1,120)	1602
51 FORMAT(1X,I4,A4,1X,120A1)	1603
WRITE(6,20)((POINTF(I,J),I=1,120),J=2,57)	1604
J=-X	1605
WRITE(6,51)J,LET,(POINTF(1,58),I=1,120)	1606
20 FORMAT(10X,120A1)	1607
WRITE(6,30)ALIM	1608
RETURN	1609
END	1610
C	1611
C	1612
SUBROUTINE LINEAR(X,LIMITY,Y)	1613
DIMENSION ENCRMT(59)	1614
LIMITY=0	1615
FLIMIT=X*2.	1616
DELTA=FLIMIT/58.	1617

ENCRMT(1)=X	1618
DO 10 I=2,58	1619
10 ENCRMT(I)=ENCRMT(I-1)-DELTA	1620
ENCRMT(59)=-X	1621
DO 11 I=1,58	1622
11 IF(Y.LE.ENCRMT(I).AND.Y.GE.ENCRMT(I+1))GO TO 20	1623
GO TO 40	1624
20 LIMITY=I	1625
40 RETURN	1626
END	1627
C	1628
C	1629
SUBROUTINE SWEEPR(XOMEGA)	1630
DIMENSION OMEGA(17),SWEPER(17)	1631
COMMON/INFO3/OMEGA	1632
DATA (SWEPER(I),I=1,17) /C,C1,0.05,0.1,0.2,0.3,0.4,0.5,0.6,0.7,	1633
*C.8,0.9,1.0,2.0,4.0,6.0,8.0,10.0/	1634
DO 70 K=1,17	1635
70 OMEGA(K)= SWEPER(K)*XOMEGA	1636
RETURN	1637
END	1638
C	1639
C	1640

APPENDIX B

FLOW CHART OF SOURCE DECK

```

*****
*C
*****
      I
      I
      I
*****
      DO 10 J=1,100
*****
*****
      I
      I
      I
*****
      CALL RTLOCS
*****
*****
      I
      I
      I
*****
      10 CONTINUE
*****
*****
      I
      I
      I
*****
      END
*****

```

(ENTRANCE)

I

```
.....
* C
* C
* C
* C
* SUBROUTINE RTLOCS
* DIMENSION SAVE1(100,100),SAVE2(100,100),XA(2000),XB(2000)
* DIMENSION X(100),IGA(100),IGB(100),C(100),D(100),ANS(100),
* ISAVE(100),ERA(100),AS(100),RS(100),A(100),B(100),ROOTR(100),
* 2ROOTI(100),ATK(100),CK(100),CKS(100)
* INTEGER CMENT(20),DENSE,ROTLCS,EXPND,FREQSR,STAR,STAR1,STAR2,
* STAR3,ONE,ZERO,DOT,VEE,AVE,DASH,BLANK,Q
* POINT(130,100),SPOT(130,50)
* DATA ONE/1H1/,ZERO/1M0/,DOT/1H./,AVE/1HA/,VEE/1HV/,BLANK/1H /,
* CASH/1H-/,Q/1H1/,STAR1/1M0/,STAR2/1M0/,STAR3/1H./
* LOGICAL PRNT
* EQUIVALENCE (SPOT(1,1),POINT(1,1)),(STAR1,ZERO),(STAR3,DOT),
* (JACKIE,KJ,1),(JOANN,KK,J)
* COMMON/FREEK/A,B
* COMMON /INFO4/ Q,BLANK,DASH
* COMMON /INFO7/ ICOUNT
* COMMON SAVE1,SAVE2,POINT
* COMMON /INFO8/ROTLCS,EXPND,FREQSR
* COMMON /INFO9/ DENSE
* READ (5,105) ROTLCS,EXPND,FREQSR,DENSE
* READ(5,11) CMENT
* WRITE(6,13) CMENT
.....
```

I

```
.....
* IF(ROTLCS.NE.0.AND.EXPND.NE.0.AND.FREQSR.NE.0) GO TO 37
.....
```

I

```
.....
* IF(ROTLCS.EQ.0.AND.EXPND.EQ.0) WRITE(6,30)
* IF(FREQSR.EQ.0.AND.EXPND.EQ.0) WRITE(6,31)
* IF(FREQSR.EQ.0.AND.ROTLCS.EQ.0) WRITE(6,32)
* IF(ROTLCS.NE.0.AND.EXPND.NE.0) WRITE(6,33)
* IF(FREQSR.NE.0.AND.EXPND.NE.0) WRITE(6,34)
* IF(FREQSR.NE.0.AND.ROTLCS.NE.0) WRITE(6,35)
.....
```

I

37

18 D0277 10U=1,2C00

I

.....
* XA(10U)=0.0
.....

.....
* 277 XB(10U)=0.0
.....

.....
* DO 401 KK=1,100
.....

.....
* DO 401 KJ=1,130
.....

.....
* 401 POINT(KJ, KK)=BLANK
.....

.....
* DO 402 KK=1,130
.....

.....
* POINT(KK, 51)=DASH
.....

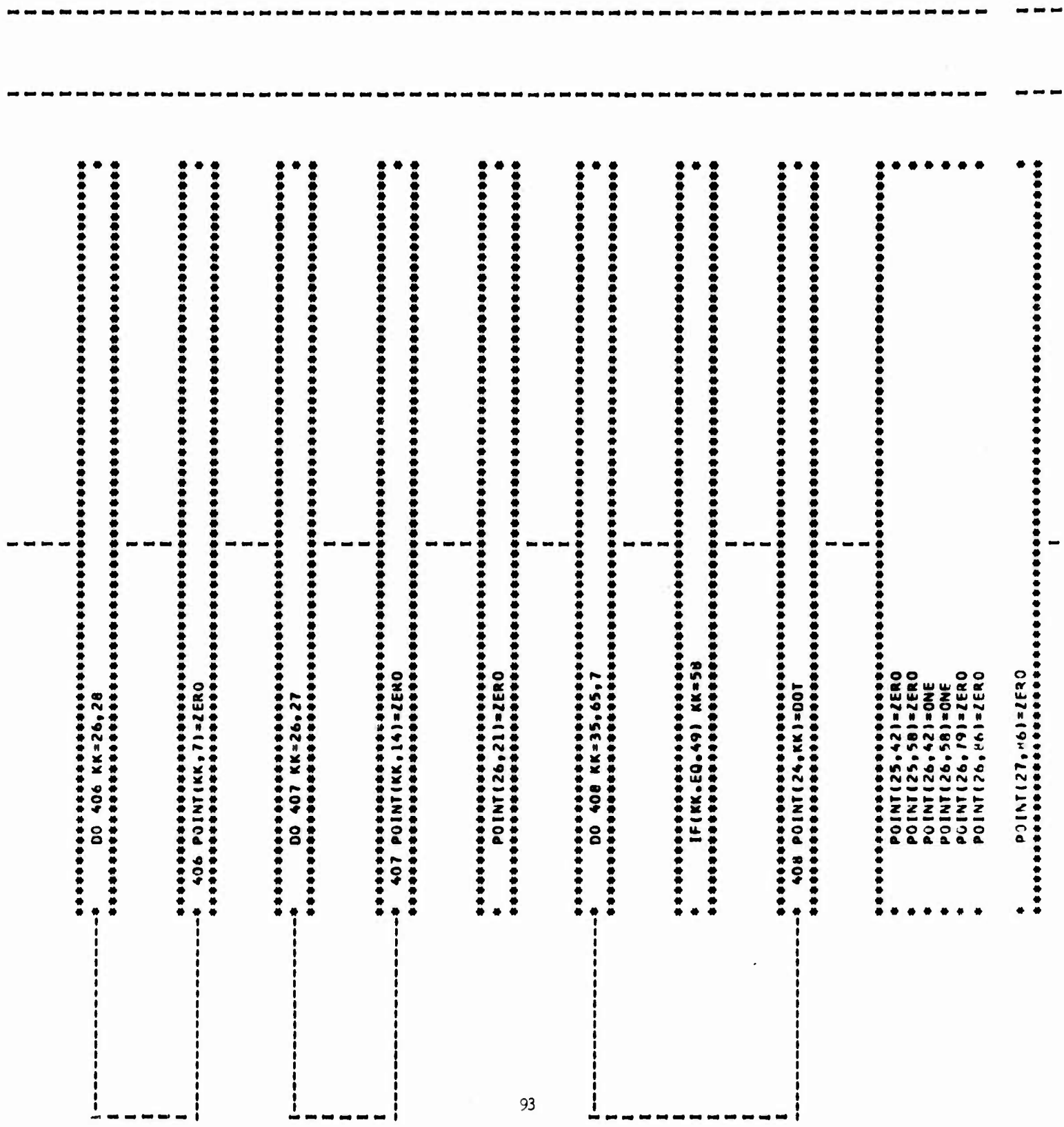
.....
* 402 POINT(KK, 50)=DASH
.....

.....
* DO 403 KK=1,100
.....

```

*****
* DO 403 KJ=34,125,13
*****
*****
* 403 POINT(KJ,KK)=Q
*****
*****
* DO 404 KJ=34,124
*****
*****
* DO 404 KK=7, 93,7
*****
*****
* IF(KK.EQ.49) KK=58
*****
*****
* 404 POINT(KJ,KK)=DASH
*****
*****
* DO 405 KK=7, 93,7
*****
*****
* IF(KK.EQ.49) KK=58
*****
*****
* 405 POINT(25,KK)=ONE
*****

```



.....
DO 409 KK=26,28
.....

.....
409 POINT(KK,43)=ZERO
.....

.....
DO 420 KK=1,100
.....

.....
IF(KK.EQ.4) KK=98
.....

.....
420 POINT(26, KK)=0
.....

.....
POINT(26,4)=VEE
POINT(26,97)=AVE
.....

.....
DO 7601 JACKIE=1,100
.....

.....
DO 7601 JCANN=1,100
.....

.....
SAVE1(JACKIE,JOANN)=0.0
.....


```

*****
* 10 WRITE(6,1)
*****

*****
DO 5 I = 1,100
*****

*****
* A(I) = 0.0
*****

*****
* 5 B(I) = 0.0
*****

*****
* FORK1=0.0
* FORK2 = 0.0
* FORK3 = 0.0
* WRITE (6,102)IPROB
* 102 FORMAT (1H1, 9X,40HPOLYNOMIAL MULTIPLICATION AND ROOT LOCUS,44X,11H
* 1HPROBLEM NO.,15//)
*****

*****
* IF (N) 21, 20, 21
*****

*****
* 20 READ(5,103)Y,DY,YT
* WRITE(6,140)N,IA,IB,IPROB
* WRITE(6,141)Y,DY,YT
* 141 FORMAT(1H0,5X,10HK-INITIAL=F20.10,5X,12HINCREMENT K=F20.10,5X,12H
* K-TERMINATE=F20.10)
* 103 FORMAT (4F10.0)
*****

```

```

*****
* GO TO 25
*****

*****
* 21 READ(5,104)(X(I),I=1,N)
* WRITE(6,142) (I,X(I),I=1,N)
* 104 FORMAT(7E10.0)
*****

*****
* 25 READ (5,105)(IGA(I), I = 1, IA)
* WRITE(6,143) (I,IGA(I),I=1,IA)
* 143 FORMAT(10X,24NUMBER OF POLY. IN GROUP, 13,14H OF NUMERATOR=,110)
* 142 FORMAT(10X,2HX(,13,2H)=,F20.10)
* READ (5,105)(IGB(J), J= 1, IB)
* WRITE(6,144) (J,IGB(J),J=1,IB)
* 144 FORMAT(10X,24NUMBER OF POLY. IN GROUP, 13,16H OF DENOMINATOR=,110)
* *)
* 105 FORMAT(7I10)
*****

*****
* 300 00 15 I = 1,100
*****

*****
* 15 SAVE(I) = 0.0
*****

*****
* ISDEG = 0
* JJ = 1
*****

*****
* 200 REAC (5,106)MDEG,( C(I), I=1, 6)
*****

```

```

*****
* IF(MDEG.GT.6) GO TO 48
*****

*****
* WRITE(6,145) (I,C(I),I=1,MDEG)
*****

*****
* IF (MDEG - 6) 49, 49, 48
*****
OK-----

*****
* 48 READ (5,104)(C(I), I = 7, MDEG)
* WRITE(6,145) (I,C(I),I=1,MDEG)
* 106 FORMAT(11C,6E10.0)
*****
OK-----

*****
* 49 IF (ICA(JJ)-1) 50, 51, 50
*****

*****
* 51 DO 56 I= 1, MDEG
*****

*****
* 56 ANS(I) = C(I)
*****

*****
* IADEG = MDEG
*****

```

```

.....
* IF (ISDEG - MDEG) 52, 52, 53
.....
.....
* 53 INDEG = ISDEG
.....
.....
* GO TO 68
.....
.....
* 52 INDEG = MDEG
.....
.....
* GO TO 68
.....
.....
* 50 READ (5,106)NDEG, (D(1), I=1, 6)
.....
.....
* IF(NDEG.GT.6) GO TO 54
.....
.....
* WRITE(6,145) (I,D(1),I=1,NDEG)
.....
.....
* IF (NDEG - 6) 55, 55, 54
.....

```

```
*****  
      * 54 READ(5,104)(D(I), I=7,NDEG)  
      * WRITE(6,145) (I,D(I), I=1,NDEG)  
*****  
*****  
      * 55 IADEG = NDEG + MDEG -1  
      * CALL POLMPY(C,MDEG,D,NDEG,ANS)  
      * IGA(JJ) = IGA(JJ) -1  
*****  
*****  
      * IF (IGA(JJ) -1) 65, 65, 64  
*****  
*****  
      * DO 60 I = 1, IADEG  
*****  
*****  
      * C(I) = ANS(I)  
*****  
*****  
      * MDEG = IADEG  
*****  
*****  
      * GO TO 50  
*****  
*****  
      * 65 IF (ISDEG - IADEG) 66,66,67  
*****
```

101

.....
* 6003 JJ = JJ + 1
.....

.....
DO 70 I = 1, INDEG
.....

.....
70 SAVE(I) = ERA(I)
.....

.....
* ISDEG = INDEG
* IA = IA - 1
.....

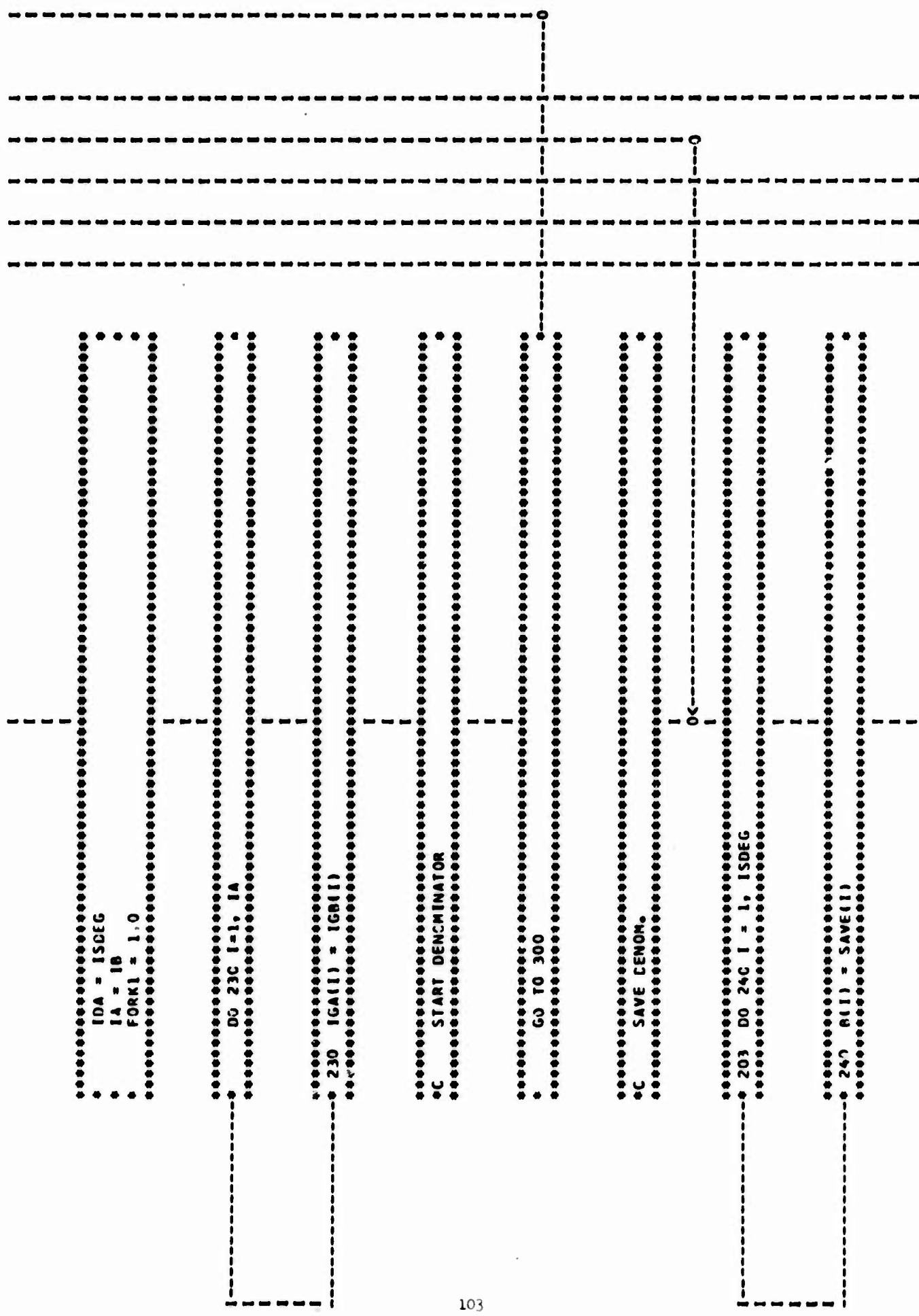
.....
* IF (IA) 201, 201, 200
.....

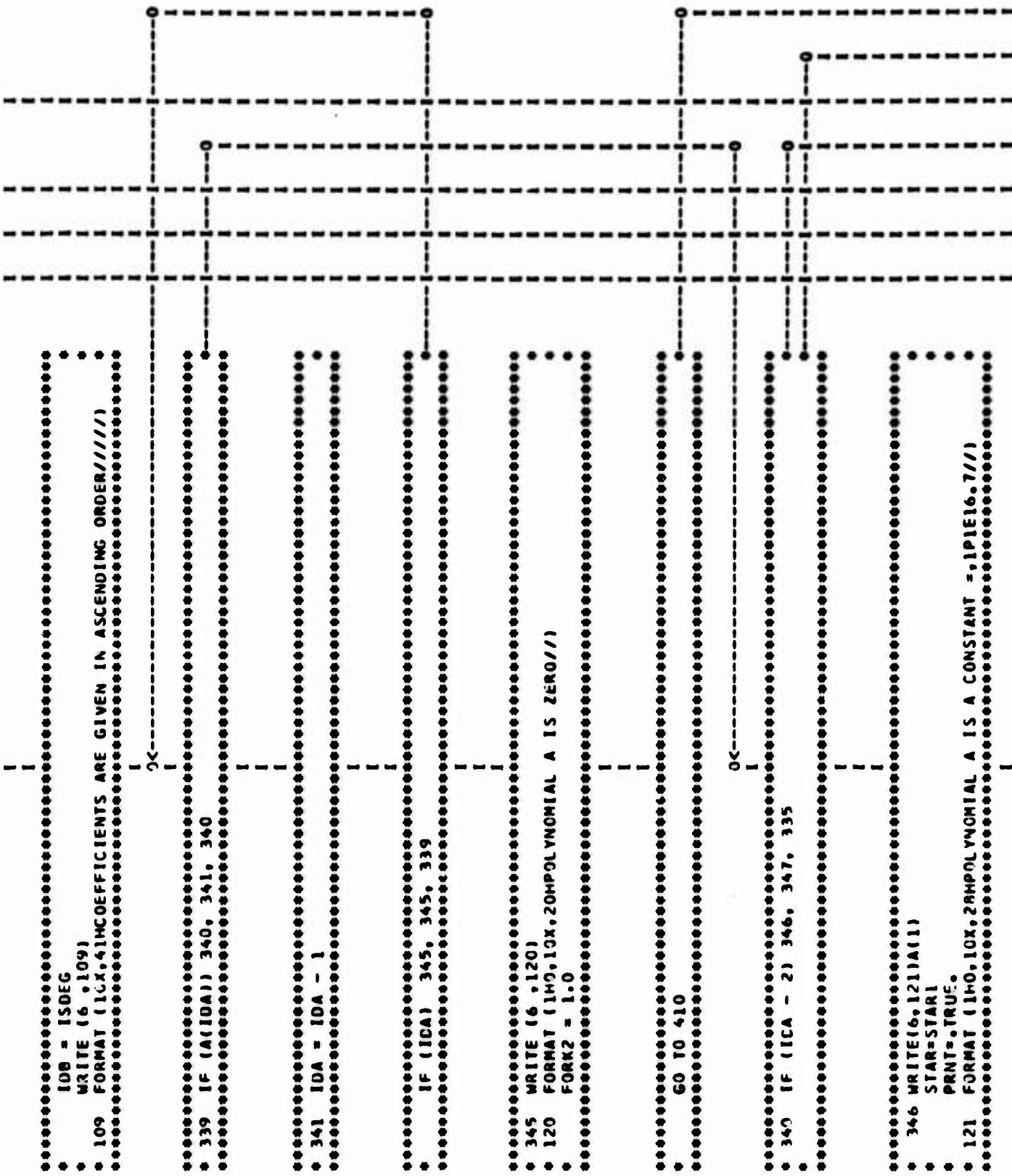
.....
* 201 IF (FCNK1) 202, 202, 203
.....

.....
*C SAVE NUMERATOR.
.....

.....
207 DO 22C I = 1, ISDEG
.....

.....
220 A(I) = SAVE(I)
.....





```

.....
*      GJ TO 41C
.....

*      OK-----0
.....

*      347  ROOT = - A(1) / A(2)
*      WRITE (6,133)A(1), A(2)
*      133  FORMAT (10X,21HTHE COEFFICIENTS OF A/1P2E2C.7)
*      WRITE (6,122)ROOT
*      STAR=STAR1
*      PRNT=.TRUE.
*      ANUMB1=ROOT
*      ANUMB2=0.3
*      CALL PLOT(SAVE1,SAVE2,ANUMB1,ANUMB2,PCINT,XA,XB,MI,PRNT,STAR,111,
*      *,NO)
*      122  FORMAT (10X,10X,23HROOT OF POLYNOMIAL A IS,1P1E16.7//)
.....

*      GJ TO 41C
.....

*      OK-----0
.....

*      WRITE POLYS
.....

*      335  IUIA =IDA -1
*      WRITE (6,107)IDIA,(A(1),I=1,IDA)
*      K = ICA
.....

*      DO 80C I = 1,IDA
.....

*      AS(I) = A(K)
.....

```

```

      800 K = K-1

```

```

      IDP2A=IDA *2
      ID2A= 2 *ID1A
      CALL MULLER (AS,ID1A,ROOTR,ROOTI)

```

```

      DO 805 I = 1, ID1A

```

```

          SAM = 100. * AMAX1(ABS(ROOTR(I)),ABS(ROOTI(I)))
          IF (SAM * ABS(ROOTR(I)).EQ. SAM) ROOTR(I)= 0.0
          IF (SAM * ABS(ROOTI(I)).EQ. SAM) ROOTI(I)= 0.0

```

```

      805 CONTINUE

```

```

      400 WRITE (6,111) (ROOTR(I),ROOTI(I),I=1,ID1A)
      CALL ERCHER(ROOTI,ID1A)
      PRINT=TRUE.
      STAR=STAR1

```

```

      DO 2 I11=1,ID1A

```

```

          ANUMB1=ROOTR(I11)
          ANUMB2=ABS(ROOTI(I11))
          CALL PLOT1R(SAVE1,SAVE2,ANUMB1,ANUMB2,PCINT,XA,XB,MI,PRNT,STAR,I11,
          *NO)

```

```

.....
2 CONTINUE
.....
* 410 IF(B(I0B)) 411, 412, 411
.....
* 412 IDA = IDA - 1
.....
* IF (ICB) 445, 445, 410
.....
* 445 WRITE (6, 123)
* 123 FORMAT (1H0,10X,20HPOLYNOMIAL B IS ZERO//)
.....
* IF (FCRK2) 12, 450, 12
.....
* 450 FORK3 = 1.0
.....
* GO TO 698
.....
* 411 IF (ICB - 2) 451, 452, 499
.....

```

```

*****
* 451 WRITE (6,124)B(108)
*      STAR=STAR2
*      PRNT=.FALSE.
*      STAR=STAR3
* 124 FORMAT (1H0,10X,28HPOLYNOMIAL B IS A CONSTANT =,1P1E16.7//)
*****

```

```

*****
*      GO TO 698
*****

```

0<-----

```

*****
* 452 ROOT = -B(1) / B(2)
*      WRITE (6,134)B(1), B(2)
* 134 FORMAT (1X,21HTHE COEFFICIENTS OF B/1P2E2C.7)
*      WRITE (6,125)ROOT
*      STAR=STAR2
*      PRNT=.TRUE.
*      ANUMB1=ROOT
*      ANUMB2=0.C
*      CALL PLOTTER(SAVE1,SAVE2,ANUMB1,ANUMB2,PCINT,XA,XB,MI,PRNT,STAR,III*
*      ,NO)
*      PRNT=.FALSE.
*      STAR=STAR3
* 125 FORMAT (1H0,10X,23HROOT OF POLYNOMIAL B IS,1P1E16.7//)
*****

```

```

*****
*      GO TO 698
*****

```

```

*****
* 107 FORMAT (1X,42HTHE COEFFICIENTS OF POLYNOMIAL A (ORDER = 13,1H)/ (
*      11P6E2C.7))
*****

```

0<-----

```

*****
* 499 ID18 = IDR -1
*      WRITE (6,1C9)ID18,(B(1),I=1,108)
* 108 FORMAT (///10X,42HTHE COEFFICIENTS OF POLYNOMIAL B (ORDER = 13,1H
*      1)/ (1P6E2C.7))
*      K = ICB
*****

```

```

*****
DO 801 I = 1, IDB
*****

*****
* RS(I) = R(K)
*****

*****
* 801 K = K-1
*****

*****
* IDP28= IDB * 2
* ID28 = 2 * ID18
* CALL MULLER (BS, ID18, ROOTR, ROOTI)
*****

*****
DO 806 I = 1, ID18
*****

*****
* SAM = 100. * AMAX1(ABS(ROOTR(I)), ABS(ROOTI(I)))
* IF (SAM + ABS(ROOTR(I)).EQ. SAM) RCCTR(I)= 0.0
* IF (SAM + ABS(ROOTI(I)).EQ. SAM) RCOTI(I)= 0.0
*****

*****
* 806 CONTINUE
*****

*****
* 500 WRITE (6,112)(ROOTR(I),RCOTI(I),I= 1,ID18)
* CALL ERCHCK(RCOTI, ID18)
* STAR=STAR2
* PRNT=.TRUE.
*****

```



```

*****
DO 3 III=1, IDIB
*****

*****
ANUMB1=ROJTR(III)
ANUMB2=ABS(ROOTI(III))
CALL PLOT1(SAVE1,SAVE2,ANUMB1,ANUMB2,POINT,XA,XB,MI,PRNT,STAR,III,
*,NO)
*****

*****
3 CONTINUE
*****

*****
PRNT=.FALSE.
STAR=STAR1
111 FORMAT (I10,I1X,I4HTHE ROOTS OF A/ (1P1E20.7,6M +I .1P1E14.7,1P1
1E20.7,6M +I .1P1E14.7,1P1E20.7,6M +I .1P1E14.7))
112 FORMAT (I10,I1X,I4HTHE ROOTS OF B/ (1P1E20.7,6M +I .1P1E14.7,1P1
1E20.7,6M +I .1P1E14.7,1P1E20.7,6M +I .1P1E14.7))
*****

*****
698 IF (FCR2)12,699,12
*****

*****
691 WRITE (6,102)IPROB
*****

*****
699 IF (FCR3)12,699,12
*****

```

```

*****
* MSHEET = 5
* START K CALCULATIONS
*****

```

```

*****
* IF (N) 702,702,533
*****

```

```

*****
* 533 DO 55C I= 1, N
*****

```

```

*****
* DO 541 J= 1, IDA
*****

```

```

*****
* 541 ATK(J) = X(I) * A(J)
*****

```

```

*****
*C COMPUTE RCOTS OF K * A + B
* IDC= MAX(1, IDA, IDB)
* CALL POLACD (ATK,IDA,B,IDB,CK)
* IDS = IDC
*****

```

```

*****
* 554 IF (CK(IDS))555, 557, 555
*****

```

```

*****
* 557 IDS = IDS - 1
*****

```

```

*****
* IF (IDS) 558,558, 554
*****

*****
* 558 WRITE (6 ,129)X(1)
* 129 FORMAT (1H0,1GX,35HPOLYNOMIAL K*A + B IS ZERO FOR K =,1P1E16.7//)
*****

*****
* GO TO 550
*****

*****
* 555 IF (IDS - 2) 559, 560, 561
*****

*****
* 559 WRITE (6 ,130)CK(IDS), X(1)
* 130 FORMAT (1H0,1GX,35HPOLYNOMIAL K*A + B IS A CONSTANT = ,1P1E15.7,10H
* 1H FOR K = ,1P1E14.7//)
*****

*****
* GO TO 550
*****

*****
* 560 ROOT = -CK(1) / CK(2)
* WRITE (6 ,131)ROOT, X(1)
* 131 FORMAT (1H0,1GX,18HROOT OF K*A + B = ,1P1E15.7,10H FOR K = ,1P1E1
* 14.7//)
*****

*****
* GO TO 550
*****

```

* 807 CONTINUE *

```

*****
*      WRITE (6,H08)IDIC,X(1),(CK(J),J=1,IDC)
*      808 FORMAT (//10X,48HTHE COEFFICIENTS OF POLYNOMIAL K+A + B (ORDER =
*      113,7H) K = 1P1E16.7/(1P6E20.7))
*      545 WRITE (6,115)(ROOTR(J),ROOTI(J),J=1,IDIC)
*      CALL ERCHER(RCOTI,IDIC)
*      CALL SAVED(RCOTR,RCOTI,IDIC,SAVE1,SAVE2,JZC,K1)
*      115 FORMAT (1H0,9X,16HRCOTS OF K+A + B/(1P1E20.7,6H + 1,1P1E14.7,1P1
*      1E20.7,6H + 1,1P1E14.7,1P1E20.7,6H + 1,1P1E14.7))
*      5452 MSHEET = MSHEET - 1
*****

*****
*      IF (MSHEET) 546, 546, 550
*****

*****
*      546 WRITE (6,102)IPROB
*      MSHEET = 5
*****

*****
*      550 CONTINUE
*****

*****
*      GO TO 12
*****

*****
*      702 DO 7C5 J = 1, IDA
*****

*****
*      705 ATK(J) = Y * A(J)
*****

```

```

*****
*C      COMPUTE ROOTS OF K * A * R
*      IDC= MAXO(IDA, IDB)
*      CALL POLADD (ATK, IDA,B, IDB,CK)
*      IDS = IDC
*****
0<-----
* 754 IF (CK(IDS))755, 757, 755
*****

*****
* 757 IDS = IDS - 1
*****

*****
* IF (IDS) 758, 758, 754
*****

*****
* 758 WRITE (6,129)Y
*****

*****
*      GO TO 711
*****
0<-----
* 755 IF (IDS - 2) 759, 760, 761
*****

*****
* 759 WRITE (6,130)CK(IDS), Y
*****

```

```

*****
*      GO TO 711
*****
0<-----0
I
I
I
*****
* 760 ROOT = -CK(1) / CK(2)
*      WRITE (6 ,131)ROOT, Y
*****
I
I
I
*****
*      GO TO 711
*****
0<-----0
I
I
I
*****
* 761 K = IDS
*****
I
I
I
*****
*      DO 804 I = 1,IDS
*****
I
I
I
*****
*      CKS(I) = CK(K)
*****
I
I
I
*****
*      804 K = K -1
*****
I
I
I
*****
*      IDIC = IDS - 1
*      IDP2C = ICS * 2
*      IDZC = 2 * IDIC
*      CALL MULLER (CKS,IDIC,ROOTR,ROOTI)
*****
I
I
I

```

```

DO 805 I = 1,IDIC
.....
      SAM = 100. * AMAX1(ABS(ROOTR(I)),ABS(ROOTI(I)))
      IF (SAM + ABS(ROOTR(I)).EQ. SAM) RCTR(I)= 0.0
      IF (SAM + ABS(ROOTI(I)).EQ. SAM) RCOTI(I)= 0.0
.....
      809 CONTINUE
.....
      WRITE (6,808) IDIC,Y,(CK(I),I=1,IDS)
      WRITE (6,115) (ROOTR(J),ROOTI(J),J=1,IDIC)
      CALL ERCMEK(ROOTI,IDIC)
      CALL SAVER(RCTR,RCOTI,IDIC,SAVE1,SAVE2,JZO,K1)
.....
      OK-----
.....
      711 Y = Y + DY
.....
      IF (Y - YT) 712,712,12
.....
      712 MSHEET = MSHEET - 1
.....
      IF (MSHEET) 713, 713, 702

```



```

.....
* 713 WRITE (6,102)IPROB
* MSHEET = 5
.....

.....
* GO TO 702
.....

0<-----0
.....
* 37 WRITE(6,36)
.....

.....
* GO TO 38
.....

0<-----0
.....
* 1016 CALL PLOTFR(SAVE1,SAVE2,ANUMB1,ANUMB2,PCINT,XA,XB,MI,PRNT,STAR,III
* ,NO)
.....

.....
* 1018 IF(EXPD.EQ.0) GO TO 1017
.....

.....
* READ(5,22)I
* 22 FORMAT(11U)
.....

.....
* IF(L.EQ.0) GO TO 1017
.....
>

```


(ENTRANCE)

```
.....
* C .....
* C .....
* C .....
* C .....
* SUBROUTINE POLMPY (A,N,B,M,C) .....
* DIMENSION A(1),B(1),C(1) .....
* K = M*N .....
.....
```

```
.....
DO 5 I=1,K .....
.....
```

```
.....
5 C(I) = 0.0 .....
.....
```

```
.....
DO 10 I=1,N .....
.....
```

```
.....
L = I-1 .....
.....
```

```
.....
DO 10 J=1,M .....
.....
```

```
.....
L = L+1 .....
.....
```

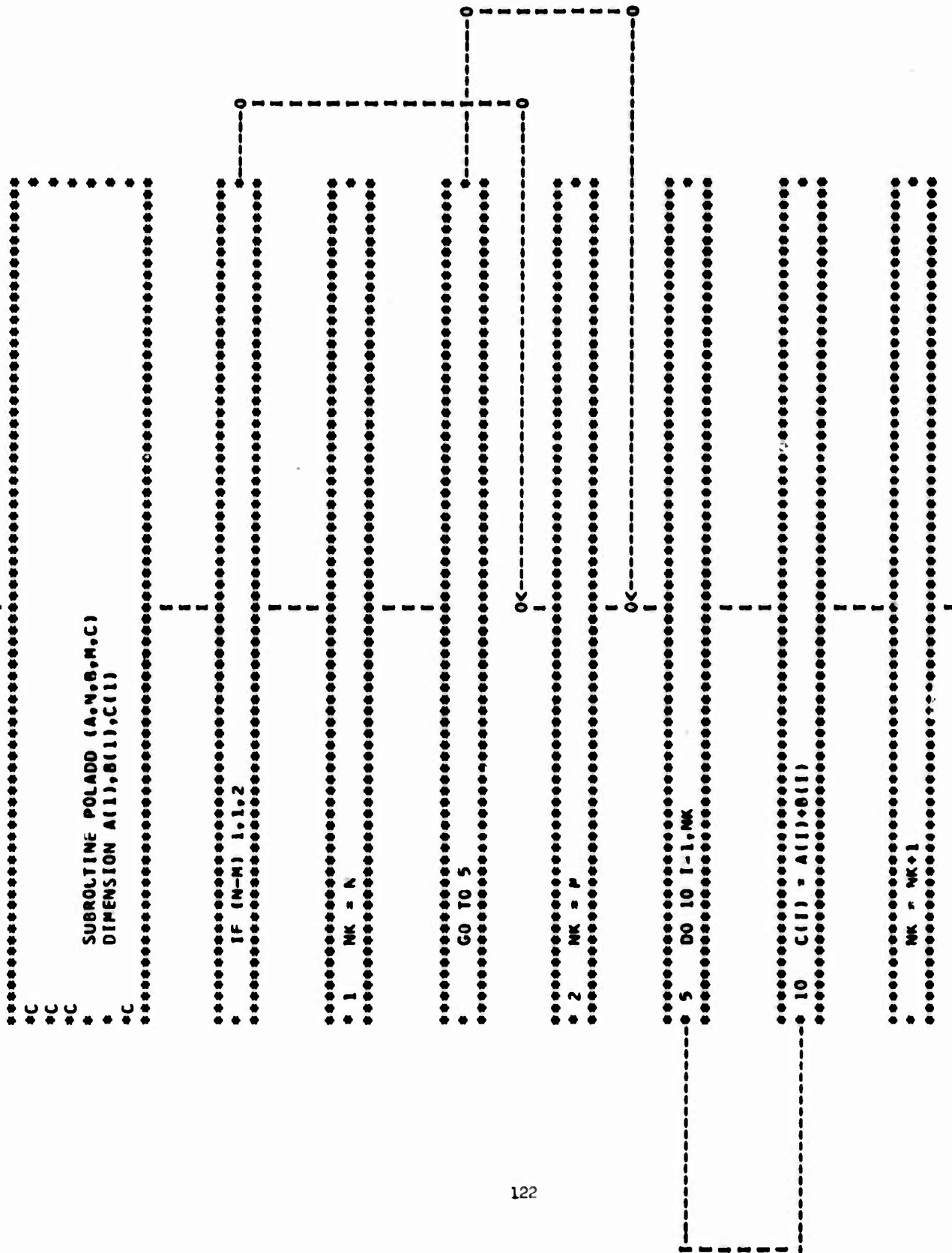
```
.....
10 C(L) = C(L)+A(I)*B(J) .....
.....
```

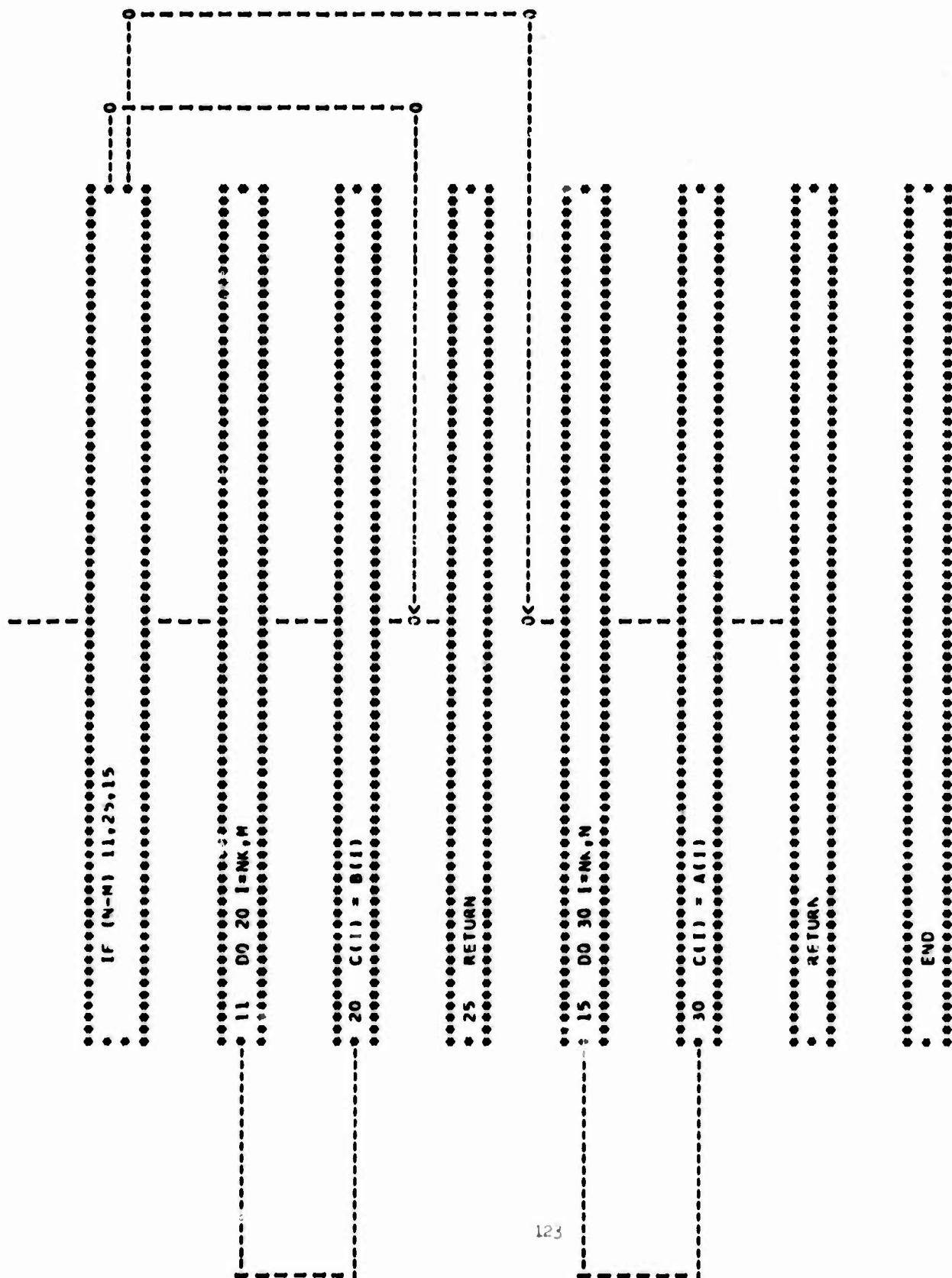
111

* RETURN *

* END *

(ENTRANCE)

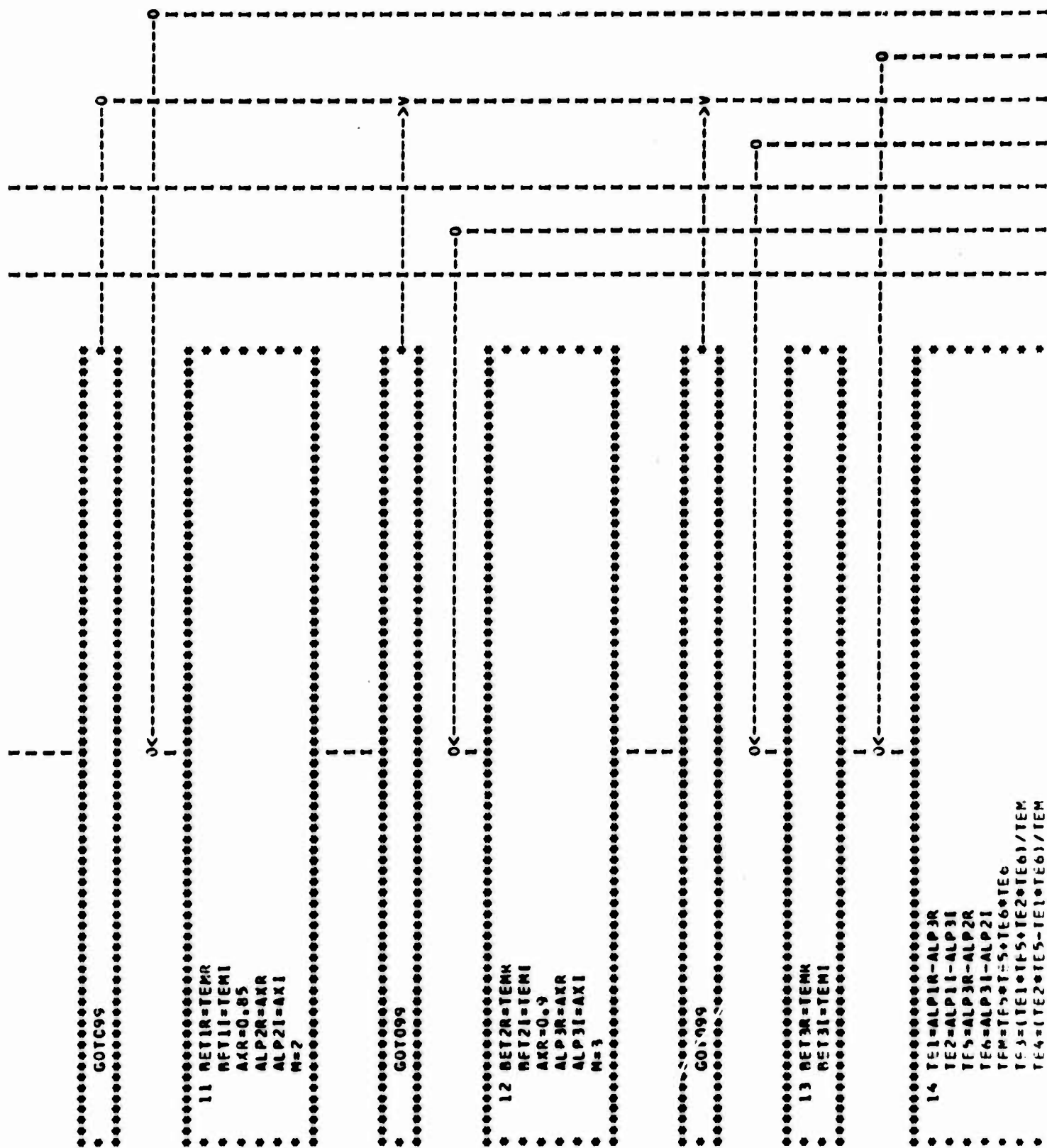




```

SUBROUTINE MULLER(COE,M1,ROOTR,ROOTI)
  DIMENSION COE(1),RCOTR(1),ROOTI(1)
  N2=M1+1
  N4=C
  I=N1+1
  10<-----
  19 IF(COE(I))9,7,9
  7 N4=N4+1
  ROOTR(N4)=0.
  ROOTI(N4)=0.
  I=I-1
  11<-----
  IF(N4-M1)19,37,19
  9 CONTINUE
  10 AXR=0.8
  AXI=C.
  L=1
  N3=1
  ALPIR=AXR
  ALPII=AXI
  M=1

```




```

*
* TE7=TE3+1.
* TE9=TE3+TE3-TE4+TE4
* TE10=2.0*TE3+TE4
* DE15=TE7+MET3R-TE4+MET3I
* DE16=TE7+MET3I+TE4+MET3R
* TE11=TE3+MET2R-TE4+MET2I+MET1R-DE15
* TE12=TE3+MET2I+TE4+MET2R+MET1I-DE16
* TE7=TE9-1.
* TE1=TE9+MET2R-TE10+MET2I
* TE2=TE9+MET2I+TE10+MET2R
* TE13=TE1-MET1R-TE7+MET3R+TE10+MET3I
* TE14=TE2-MET1I-TE7+MET3I-TE10+MET3R
* TE15=DE15+TE3-DE16+TE4
* TE16=DE15+TE4+DE16+TE3
* TE1=TE13+TE13-TE14+TE14-4.0*(TE11+TE15-TE12+TE16)
* TE2=2.0*TE13+TE14-4.0*(TE12+TE15+TE11+TE16)
* TEM = SQRT (TE1+TE2+TE2)
*
*
* IF(TE1)113,113,112
*
*
*
*
* 113 TE4 = SQRT (.5 * (TEM - TE1))
* TE3=.5*TE2/TE4
*
*
*
* GO TO 111
*
*
* OK
*
* 112 TE3 = SQRT (.5 * (TEM + TE1))
*
*
*
* IF(TE2)11C,2CG,200
*
*
*
* 110 TE3=-TE3
*
*
* JK

```

```

.....
* 230 TE4=5*TE2/TE3
.....
      OK-----
.....
* 111 TE7=TE13+TE3
*   TE8=TE14+TE4
*   TE9=TE13-TE3
*   TE10=TE14-TE4
*   TE1=2*TE15
*   TE2=2*TE16
.....

* IF (TE7+TE8+TE9+TE10+TE11+TE12+TE13+TE14+TE15+TE16)
*   IF (TE7+TE8+TE9+TE10+TE11+TE12+TE13+TE14+TE15+TE16)
.....

* 204 TE7=TE9
*   TE8=TE10
.....
      OK-----
.....

* 205 TEM=TE7+TE7+TE8+TE8
*   TE3=(TE1+TE7+TE2+TE8)/TEM
*   TE4=(TE2+TE7-TE1+TE8)/TEM
*   AXR=ALP3R+TE3+TE5-TE4+TE6
*   AXI=ALP3I+TE3+TE6+TE4+TE5
*   ALP4R=AXR
*   ALP4I=AXI
*   M=4
.....

* GO TO 99
.....
      OK-----
.....

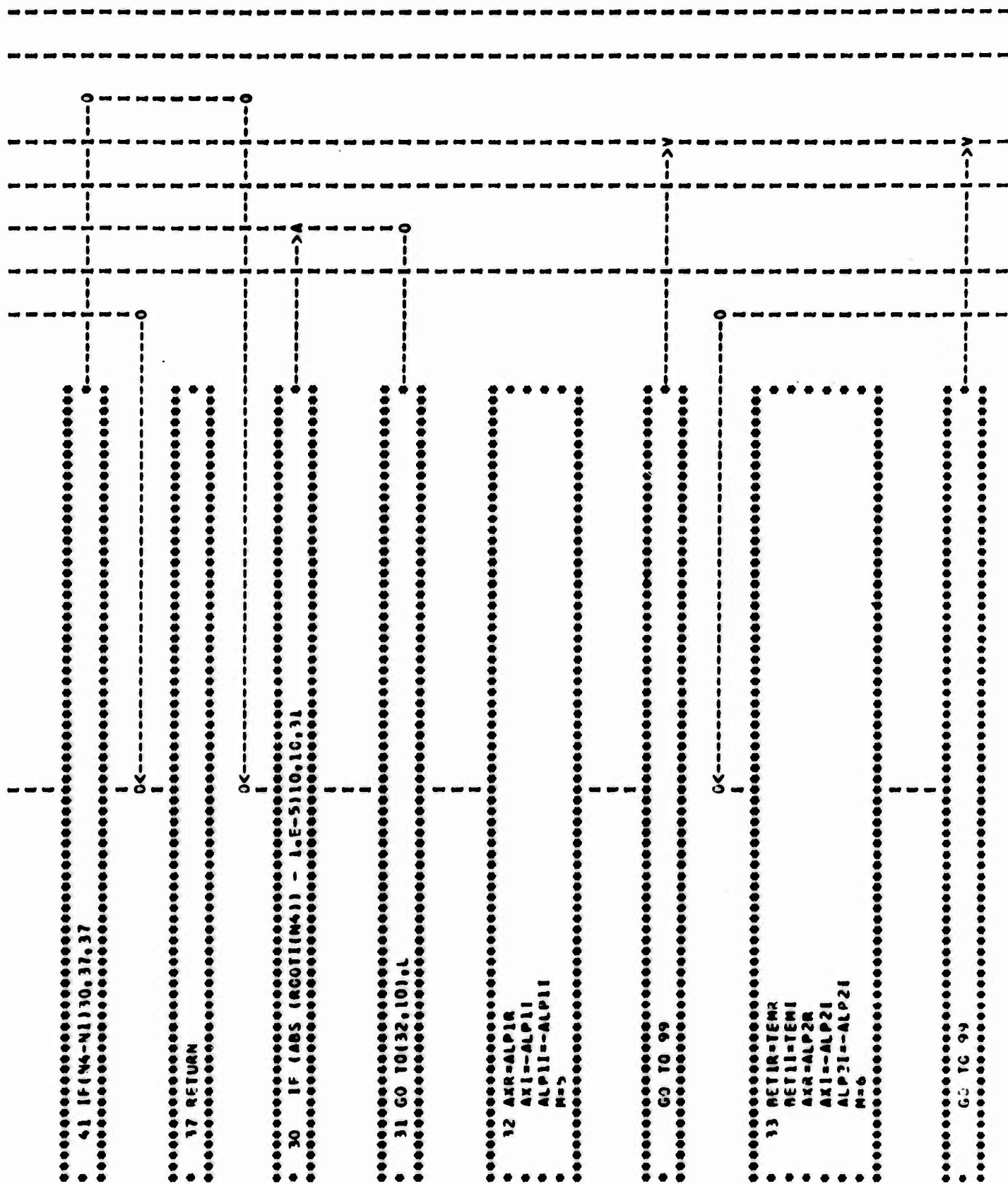
* 15 '46=1
.....

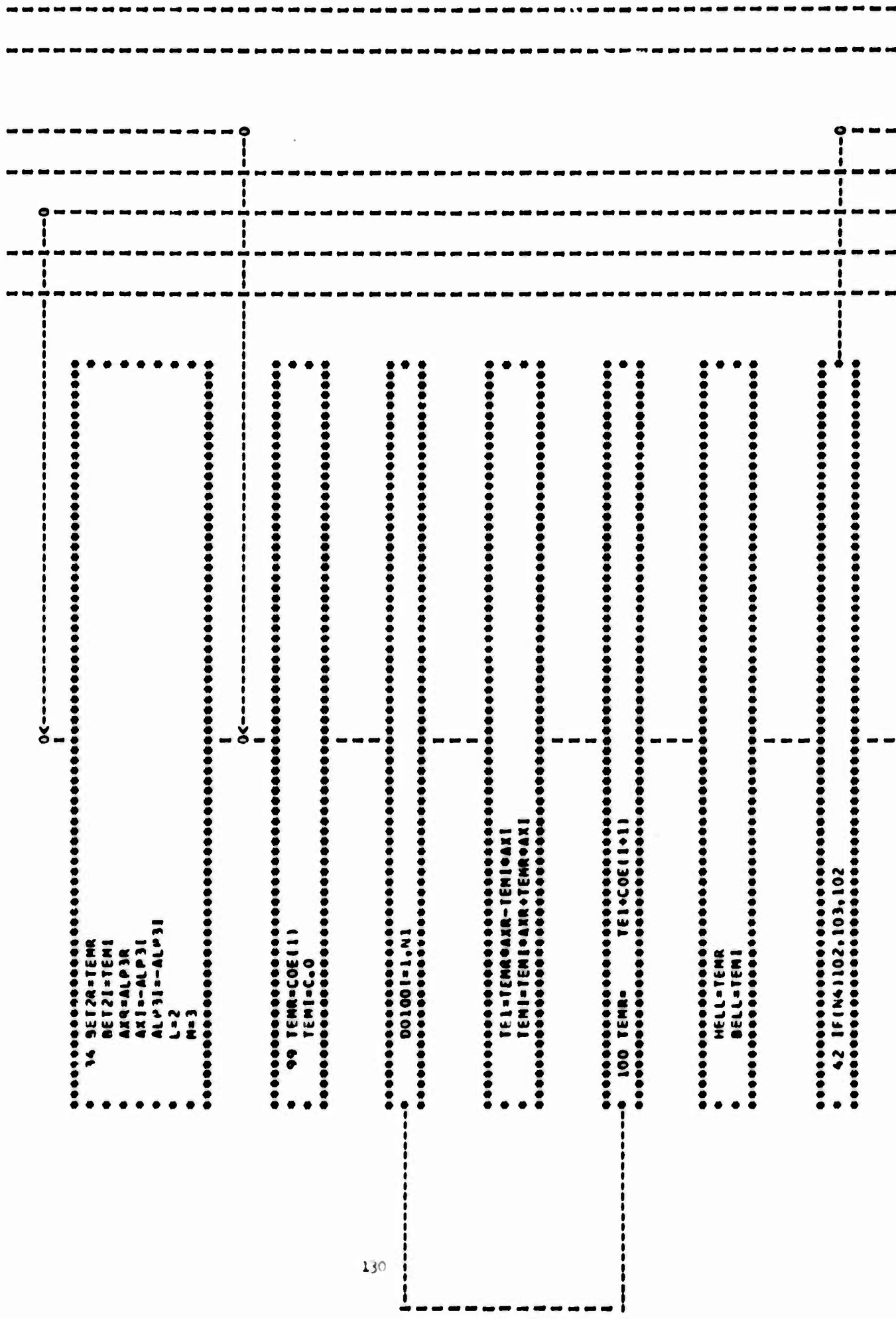
```

```

.....
* 38 IF (ABS (MELL) + ABS (BELL) - 1.E-20) 18,18,16
.....
.....
* 16 TE7 = ABS (ALP3R - AXR) + ABS (ALP3I - AXI)
.....
.....
* IF (TE7 / (ABS (AXR) + ABS (AXI)) - 1.E-7) 18,18,17
.....
.....
* 17 N3=N3+1
* ALP1R=ALP2R
* ALP1I=ALP2I
* ALP2R=ALP4R
* ALP2I=ALP3I
* ALP3R=ALP4R
* ALP3I=ALP4I
* BET1R=BET2R
* BET1I=BET2I
* BET2R=BET3R
* BET2I=BET3I
* BET3R=TEM3
* BET3I=TEM1
.....
.....
* IF (N3-100) 14,18,18
.....
.....
* 18 N4=N4+1
* ROOTR(N4)=ALP4R
* ROOTI(N4)=ALP4I
* N3=0
.....

```







(ENTRANCE)

```

*****
C
C
C
C
SUBROUTINE EXPLOT(XARRAY, IDIMEN, YARRAY, KOIMEN, BLIMIT, LET)
INTEGER POINT(130,100), GRAPHL(120,50), STAR
DIMENSION XARRAY(IDIMEN), YARRAY(KOIMEN)
DIMENSION ENCRM1(120), ENCRM2(50)
DIMENSION SAVE1(100,100), SAVE2(100,100)
COMMON SAVE1, SAVE2, POINT
EQUIVALENCE(POINT(1,1), GRAPHL(1,1))
DATA STAR/147
WRITE(6,1)
READ(5,10) D1, D2
D3=BLIMIT
D4=-BLIMIT
IF(D2.GT.D1) CALL SWITCH(D1,D2)
IF(D4.GT.D3) CALL SWITCH(D3,D4)
IC1=D3
JDIMEN=KDIMEN
CALL PLANKR(GRAPHL,120,50,120)
*****

```

132

```

*****
DO 22 I=1, JOIMEN
*****
C
C
C
C
X=XARRAY(I)
Y=YARRAY(I)
*****
C
C
C
C
IF(X.EQ.0.0.AND.Y.EQ.0.0) GO TO 22
*****
C
C
C
C
CALL LINAR (X,LIMITX,120,D1,D2,ENCRM1)
CALL LINAR (Y,LIMITY,50,D3,D4,ENCRM2)
*****

```

```

.....  

IF(LIMITX.EQ.C.OH.LIMITY.FQ.O) GO TO 22  

.....  

LIMITX=I21-LIMITX  

GRAPHL(LIMITX,LIMTY)=STAR  

.....  

OK-----O  

.....  

22 CONTINUE  

.....  

WRITE(6,30) IC1,LET,(GRAPHL(I,I),I=1,120)  

IC1=D4  

WRITE(6,34)((GRAPHL(I,J),J=1,120),J=2,57)  

WRITE(6,30) IC1,LET,(GRAPHL(I,58),I=1,120)  

WRITE(6,35) DZ,D1  

FORMAT(I,M1)  

10 FORMAT(6F10.0,20X)  

30 FORMAT(IX,I4,A4,IX,120A1)  

34 FORMAT(10X,120A1)  

35 FORMAT(3X,1P1E10.3,7HRAD/SEC,  

91X,1P1E10.3,7HRAD/SEC)  

.....  

61 RETURN  

.....  

END

```


(ENTRANCE)

```
*****
* C
* C
* C
* SURROUTINE= LINAR (Y,LIMITY,LDIMEN,DIMNS1,DIMNS2,ENCRMT)
* DIMENSION ENCRMT(LDIMEN)
* DIMENS=DIMNS1-DIMNS2
* LIMITY=0
* J=LDIPEN-1
* A=J
* DELTA=DIMENS/A
* ENCRMT(1)=DIMNS1
* ENCRMT(LDIMEN)=DIMNS2
*****
```

```
*****
* DO 10 I=2,J
*****
```

```
*****
* 10 ENCRMT(I)=ENCRMT(I-1)-DELTA
*****
```

```
*****
* DO 11 I=1,J
*****
```

```
*****
* 11 IF(Y.LE.ENCRMT(I).AND.Y.GE.ENCRMT(I+1))GO TO 20
*****
```

```
*****
* GO TO 40
*****
```

```
*****
* 20 LIMITY=I
*****
```

```
*****
* 40 RETURN
*****
```

```
*****
* END
*****
```


(ENTRANCE)

```

*****
C
C
C
C
SUBROUTINE EXPAND(ILOVE,X,Y)
  DIMENSION SAVE1(100,100),SAVE2(100,100),X2(2000),Y2(2000),X(2000),
  Y(2000),X1(2000),Y1(2000),Z1(125),Z1(125)
  INTEGER SPOT(130,50),DASH,Q,BLANK,POINT(130,100)
  DATA Q/1H1/, DASH/1H-/,BLANK/1H /
  DATA APLUS/0.0/,BPLUS/0.0/,AMINUS/C.0/,RMINUS/U.0/
  COMMON /INFO4/ Q,BLANK,DASH
  COMMON /P/YNE/AM,AP,BH,RP
  COMMON /PJ/X1,Y1
  COMMON SAVE1,SAVE2,POINT
  EQUIVALENCE(AP,APLUS),(AM,AMINUS),(BH,RMINUS),(RP,BPLUS)
  EQUIVALENCE(X1(1),Z1(1)),(Y1(1),Z1(1))
  EQUIVALENCE (SPOT(1,1),POINT(1,1)),(11,12)
*****

```

136

```

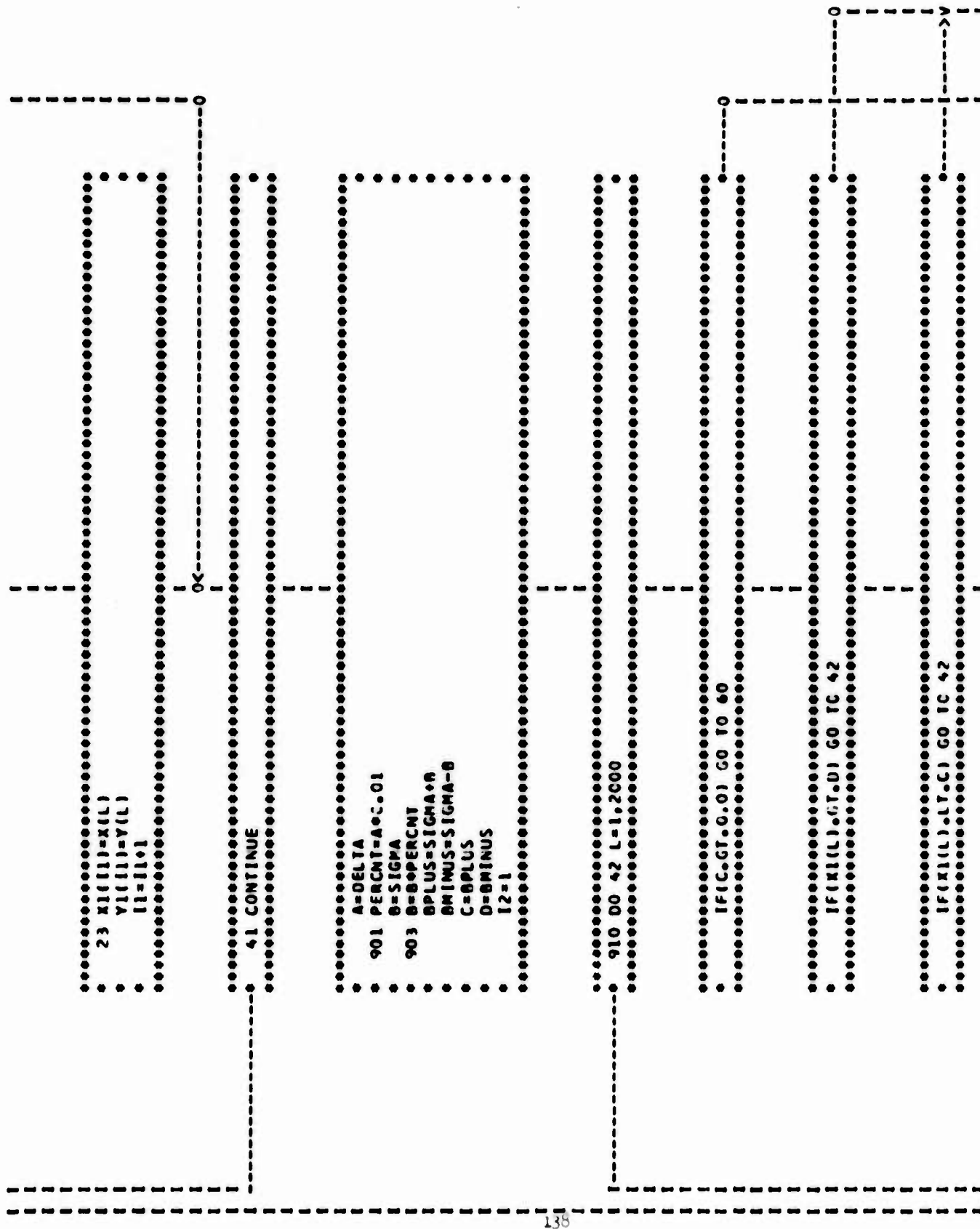
*****
DO 40 J=1,ILOVE
*****
*****
DO 50 M=1,2000
*****
*****
X1(M)=0.0
*****
*****
50 Y1(M)=0.0
*****
*****
CALL BLANKR(SPOT,130,50,126)
*****
*****

```

```

      DO 54 M=1,2000
      X2(M)=0.0
      Y2(M)=0.0
      READ(5,10) OMEGA,ENCRMT,SIGMA,DELTA
      OMEGA=ABS(OMEGA)
      10 FORMAT(4F10.0)
      A=ENCRMT
      PERCENT=A*0.01
      B=OMEGA
      B=B*PERCENT
      APLUS=B+OMEGA
      AMINUS=OMEGA-B
      C=APLUS
      D=AMINUS
      I1=1
      DO 41 L=1,2000
      IF(Y(L).LT.D)GO TO 41
      IF(Y(L).GT.C)GO TO 41

```



```

.....
GO TC 25
.....
      OK-----0
.....
      60 IF(X1(L).GT.C) GO TC 47
.....
.....
      IF(X1(L).LT.D) GO TC 42
.....
      OK-----Y
.....
      25 X2(12)=X1(L)
      Y2(12)=Y1(L)
      IZ=IZ+1
.....
      OK-----0
.....
      42 CONTINUE
.....
      CALL SPLIT(X2,Y2,SPOT,APLUS,AMINUS,BPLUS,BMINUS)
.....
      40 CONTINUE
.....
      RETURN
.....
END

```

(ENTRANCE)

```
.....
*C .....
*C .....
*C .....
*C .....
* SURROUTINE SPLIT(X,Y,SPOT,APLUS,AMINUS,BPLUS,BMINUS)
* DIMENSION X1(200),Y1(200)
* DIMENSION X(200),Y(200),Z(125),Z1(47)
* INTEGER SPOT(130,50)
* COMMON /PJ/X1,Y1
* EQUIVALENCE(X1(1),Z(1)),(Y1(1),Z1(1))
* A=APLUS
* B=AMINUS
* C=A-B
* D=BPLUS
* E=BMINUS
* G=ABS(D)
* H=ABS(E)
* F=G-H
* DELTA=C/124.0
* DIFF=F/50.0
.....
```

141

```
.....
DO 11 J=1,124
.....
```

```
.....
* Z(J)=A
.....
```

```
.....
11 A=A-DELTA
.....
```

```
.....
* Z(125)=B
* IF(C.LT.0.0)DIFF=-DIFF
.....
```

```
.....
DO 12 J=1,49
.....
```

```

.....
Z1(J)-D
.....

.....
12 D-D-CIFF
.....

.....
Z1(50)-E
CALL PRKUP(Z,X,Y,SPOT,Z1)
.....

.....
RETURN
.....

.....
E'D
.....

```


(ENTRANCE)

```
.....
* C
* C
* C
* C
* SUBROUTINE WRKUP(VV,X,Y,SPOT,XX)
* LOGICAL SKIP,SKIP1
* INTEGER SPOT(130,50),
* DATA STAR/1M0/
* DIMENSION X(200),Y(200),VY(125),X(87)
* COMMON /PAYNE/AM,AP,PM,RP
* EQUIVALENCE(AP,APLUS),(AM,AMINUS),(PM,BPLUS)
* SKIP=.FALSE.
* SKIP1=.FALSE.
.....
```

..... DC 43 J=1,2000

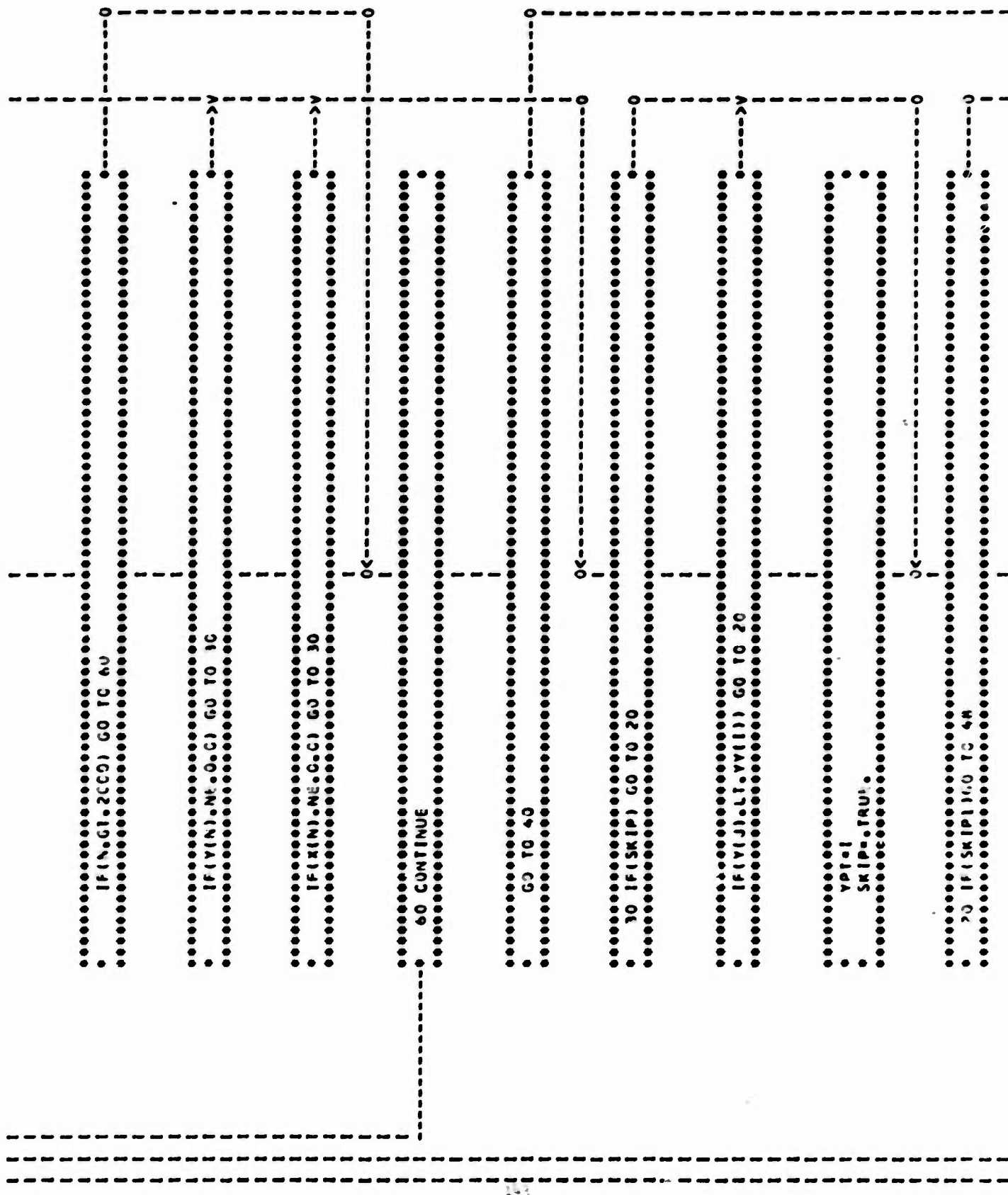
..... DO 43 I=1,125

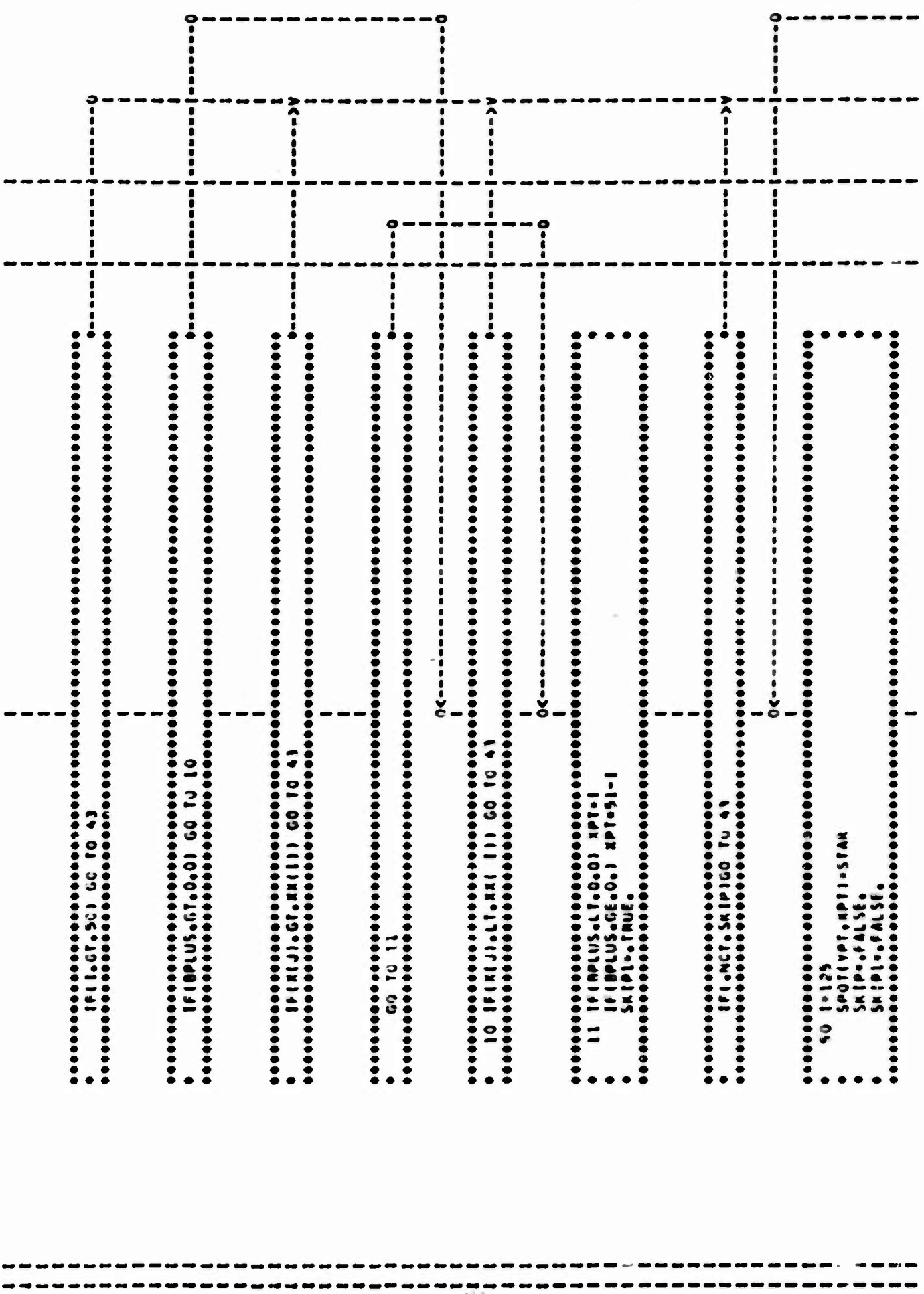
..... IF(Y(J),N1,0,C) GO TO 30

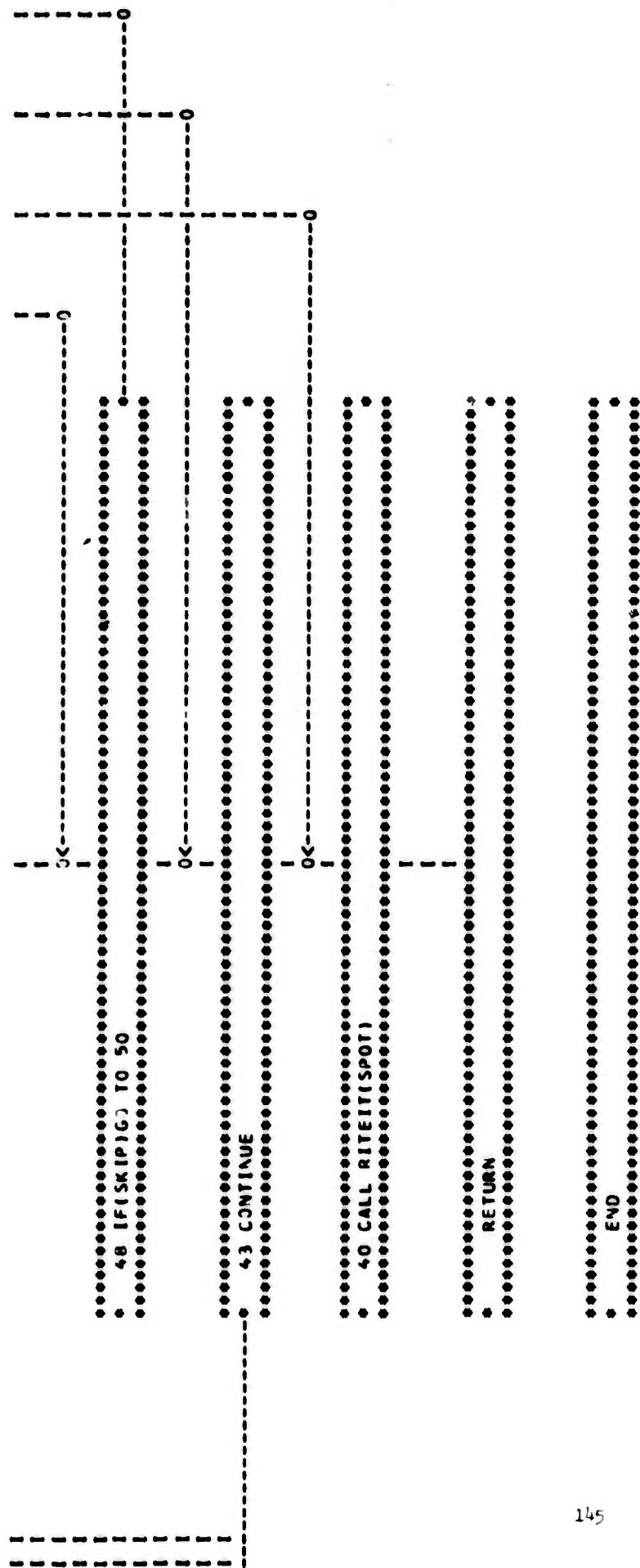
..... IF(X(J),N1,0,C) GO TO 30

..... L=J+1
..... M=J+6

..... DC 40 N=L,M







```

(ENTRANCE)
I
I
.....
C
C
C
C
.....
SUBROUTINE RITEIT(SPOT)
COMMON /PAYNE/AM,AP,PM,BP
EQUIVALENCE(AP,APLUS),(AM,AMINUS),(PM,BMINUS),(AP,BPLUS)
INTEGER SPOT(130,5C)
.....
I
I
.....
IF(BP.LT.PM) GO TO 40
.....
I
I
.....
A=BP
B=BP
WRITE(6,1)
1 FORMAT(1H1,40X,43HCOMPLEX FREQUENCY PLANE,RIGHT HAND QUADRANT )
.....
I
I
.....
GO TO 30
.....
OK-----
I
.....
40 A=BM
B=BP
WRITE(6,2)
2 FORMAT(1H1,40X,43HCOMPLEX FREQUENCY PLANE, LEFT HAND QUADRANT )
.....
I
OK-----
I
.....
30 WRITE(6,12)AP,AM
12 FORMAT(114X,15HC-----J-OMEGA,/1X,F8.2,112X,F8.2)
WRITE(6,14)B
14 FORMAT(124X,5HSIGMA,/126X,1H1,/126X,1HV,/121X,F8.2)
11 FORMAT(1X,130A1)
WRITE(6,15)A
15 FORMAT(60X,27HLINEAR EXPAND PLOT(RAD/SEC) ,34X,F8.2)
.....
I

```

.....
RETURN.....

.....
END.....

(ENTRANCE)

```

.....
* C
* C
* SUBROUTINE SAVER(RCOTR,RCOTI,IDIC,SAVE1,SAVE2,JZO,K1)
* DIMENSION SAVE1(100,100),SAVE2(100,100),ROOTR(100),ROOTI(100)
.....

```

```

.....
* IF(K1)30,9,10
.....

```

```

.....
* 9 K1=1
.....

```

```

.....
* 10 ID1=ICIC
* ID1C=ID1C+(K1-1)
.....

```

```

.....
* IF(ID1C.GE.100)GO TO 30
.....

```

```

.....
* 50 DO 40 IZAP=K1,ID1C
.....

```

```

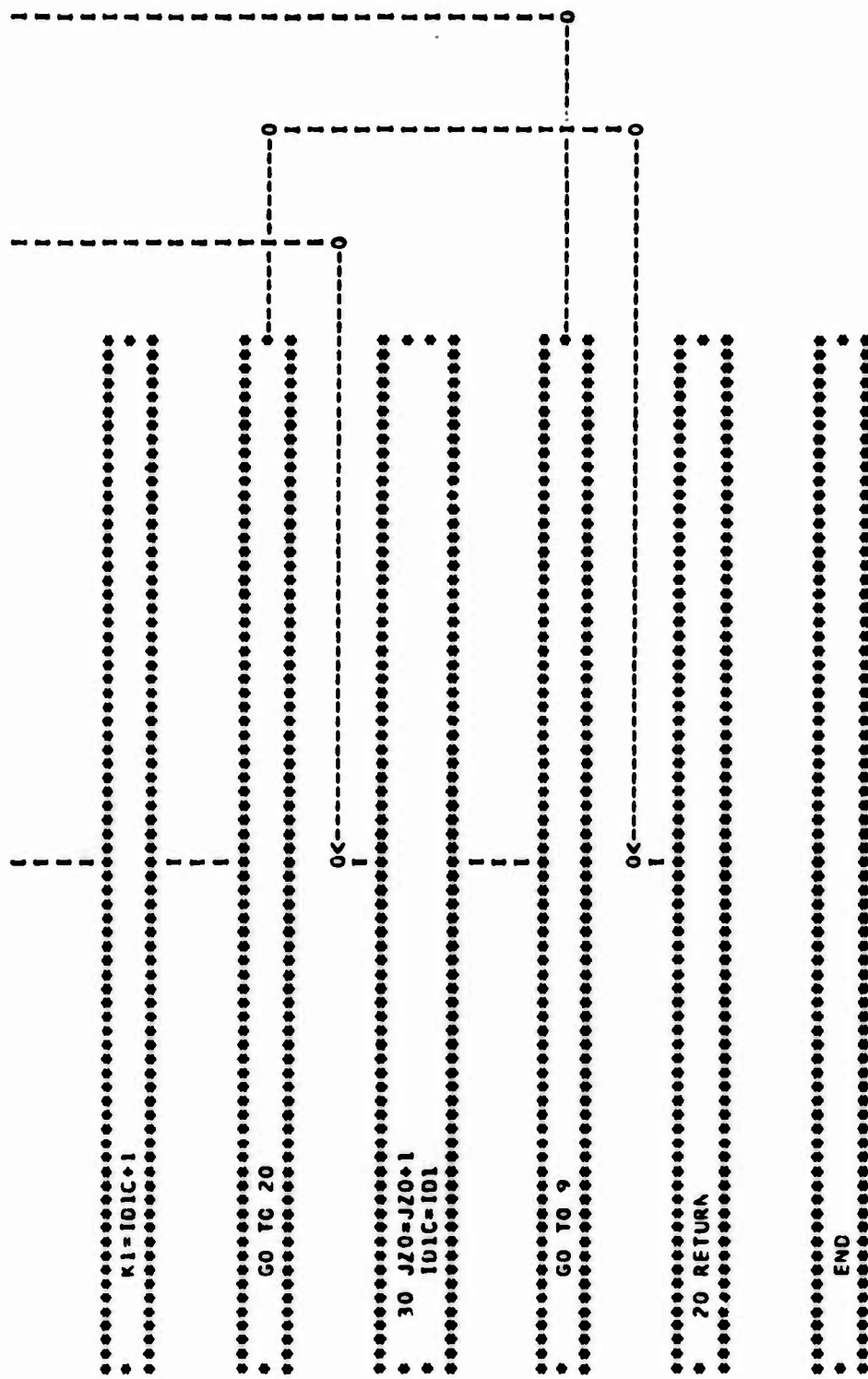
.....
* IZA=IZAP-(K1-1)
* SAVE1(JZO,IZAP)=ROOTR(IZA)
* SAVE2(JZO,IZAP)=ROOTI(IZA)
.....

```

```

.....
* 40 CONTINUE
.....

```



(ENTRANCE)

```
*****
* C
* C
* C
* C
* SURROUTINE PLOTTER(SAVE1,SAVE2,ANUMB1,ANUMB2,POINT,XA,XB,M1,PRNT,
* STAR,NAB,NO)
* DIMENSION SAVE1(100,100),SAVE2(100,100),XA(2000),XB(2000)
* INTEGER PLINT(130,100),DASH,BLANK,STAR,ROTCLS,EXPND,FREQSR,DENSE
* COMMON /INFO7/ ICOUNT
* COMMON /INFO8/ROTCLS,EXPND,FREQSR
* COMMON /INFO9/ DENSE
* LOGICAL PRNT
*****
```

```
*****
* IF(PRNT) GO TO 43
*****
```

```
*****
* DO 40 NAB=1,100
*****
```

```
*****
* DO 40 NBB=1,100
*****
```

```
*****
* IF(SAVE1(NAB,NBB))41,42,41
*****
```

```
*****
* 42 IF(SAVE2(NAB,NBB))41,40,41
*****
```

```
*****
* 41 ANUMB1=SAVE1(NAB,NBB)
* ANUMB2=SAVE2(NAB,NBB)
* ANUMB2=ABS(ANUMB2)
*****
```

```

.....
* 43 AZZ=ANUM02
*   AZZ=AZZ+10000.0
*   NZZ=AZZ
.....
.....
* IF(NZZ.EQ.0) GO TO 45
.....
.....
* 50 NBR=NBR+1
.....
.....
* 45 CALL EXECUTE(ANUM01,ANUM02,POINT,XA,XB,M1,NO,STAR)
.....
.....
* IF (CENSE.NE.1) GO TO 10
.....
.....
* ICGUNT=ICGUNT+1
.....
.....
* GO TO 11
.....
.....
* 10 IF(PRT) ICGUNT=ICGUNT+1
.....
.....

```


(ENTRANCE)

```

.....
* C
* C
* C
* C
* SUBROUTINE EXECUTE(ANUMR1,ANUMR2,POINT,XA,XH,MI,NG,STAR)
* LOGICAL SKIP1,SKIP2,LESS
* INTEGER RGTLC5,EXPND,FREQSR
* DIMENSION XA(2000),XR(2000)
* INTEGER POINT(10,100),DASH,BLANK,STAR
* COMMON /INFO8/RGTLC5,EXPND,FREQSR
* DATA KI/10/,K2/100/,K3/1000/,K4/10000/,K5/100000/,K6/1000000/,K7/10000000/
* LESS=.FALSE.
* I=0
* J=0
* XA(NG)=ANUMR1
* XR(NG)=ARS(ANUMR2)
* NG=NG+1
* IF(RGTLC5 EQ 0) RETURN
.....

```

```

.....
* IF(ABS(ANUMR1).GT.10000.0) GO TO 5C
.....

```

```

.....
* IF(ABS(ANUMR2).GT.10000.0) GO TO 5C
.....

```

```

.....
* IF(ABS(ANUMR1).EQ 0.0) GO TO 5CC
.....

```

```

.....
* IF(ABS(ANUMR2).EQ 0.0) GO TO 500
.....

```

```

.....
* IF(ABS(ANUMR2).LT 0.001) GO TO 5C
.....

```


(ENTRANCE)

```

.....
C
C
C
C
SUBROUTINE SCALE1(K1,K2,K3,K4,NEGONE,I,ANUMB1,ICONS,LESS,SKIPI)
LOGICAL LESS
LOGICAL SKIPI
SKIPI=.FALSE.
ICONS=1
I=1
AKEEP=ANUMB1
.....

```

```

.....
O<-----
I
21 NUMB1=ANUMB1
NUMB=ABS(NUMB1)
.....

```

```

.....
IF(NUMB.EQ.0)GOTO 12
.....

```

```

.....
IF(NUMB1)31,40,40
.....

```

```

.....
O<-----
I
4C IF(NUMB1.GE.10)GOTO 11
.....

```

```

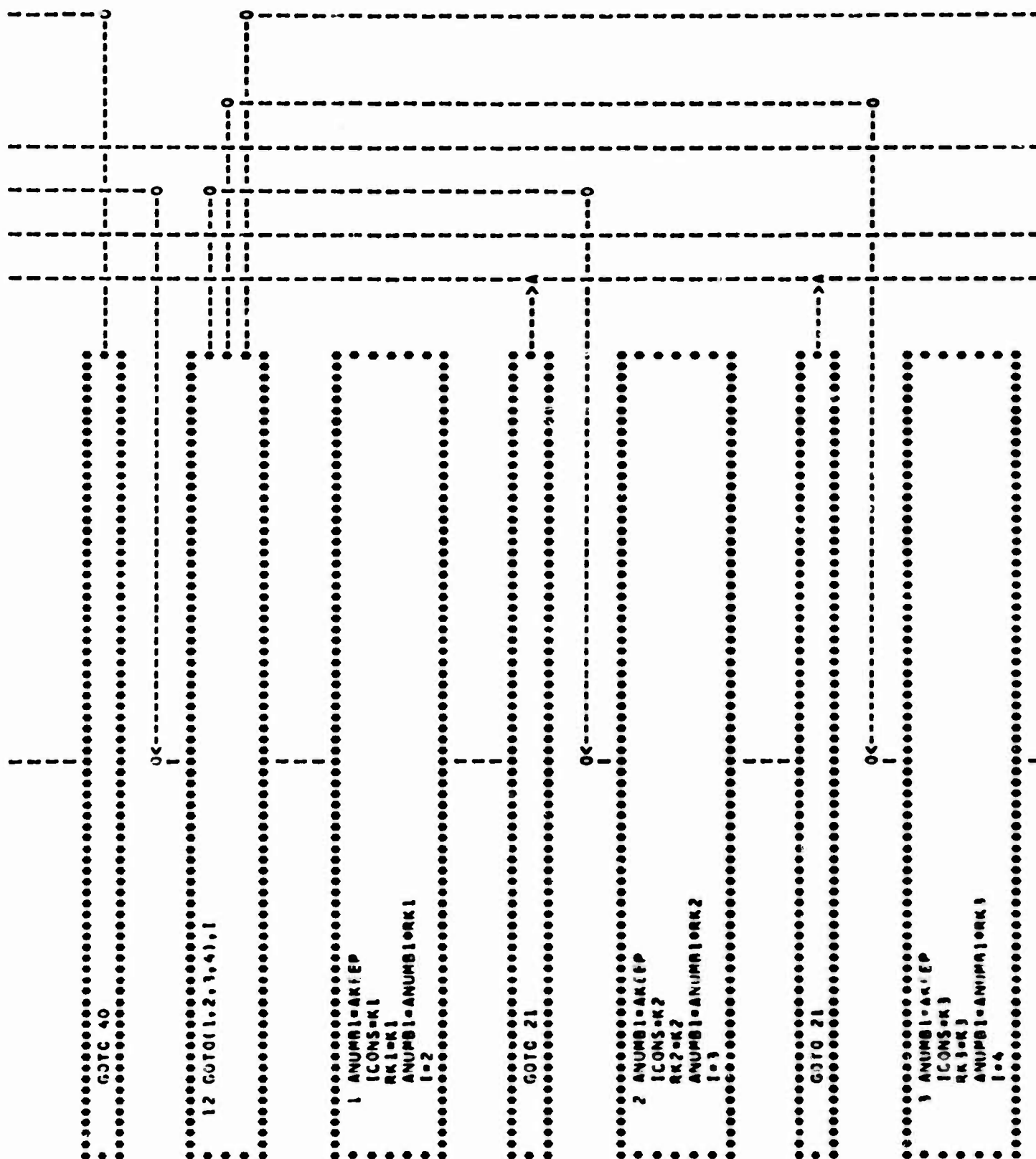
.....
GO TO 50
.....

```

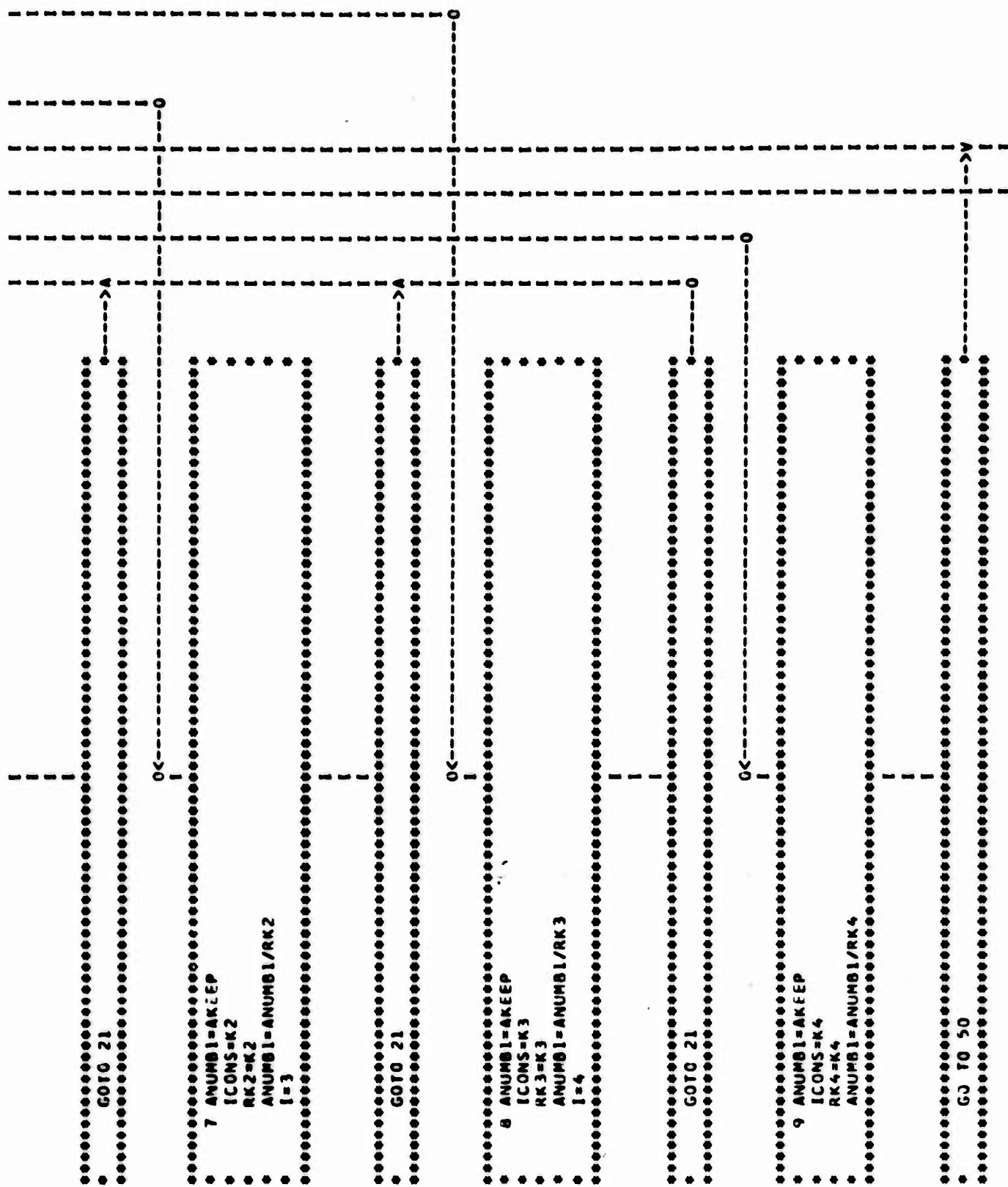
```

.....
O<-----
I
11 LESS=.TRUE.
XXX=-ANUMB1
NUMB1=XXX
.....

```



157




```

SUBROUTINE SCALE2(K1,K2,K3,K4,J,JCCNS,ANUMB2,SKIP2)
  LOGICAL SKIP2
  SKIP2=.FALSE.
  JCONS=1
  J=1
  ANUMB2=ABS(ANUMB2)
  BKEEP=ANUMB2

  20 NUMB2=ANUMB2

  IF(NUMB2.EQ.0)GO TO 10

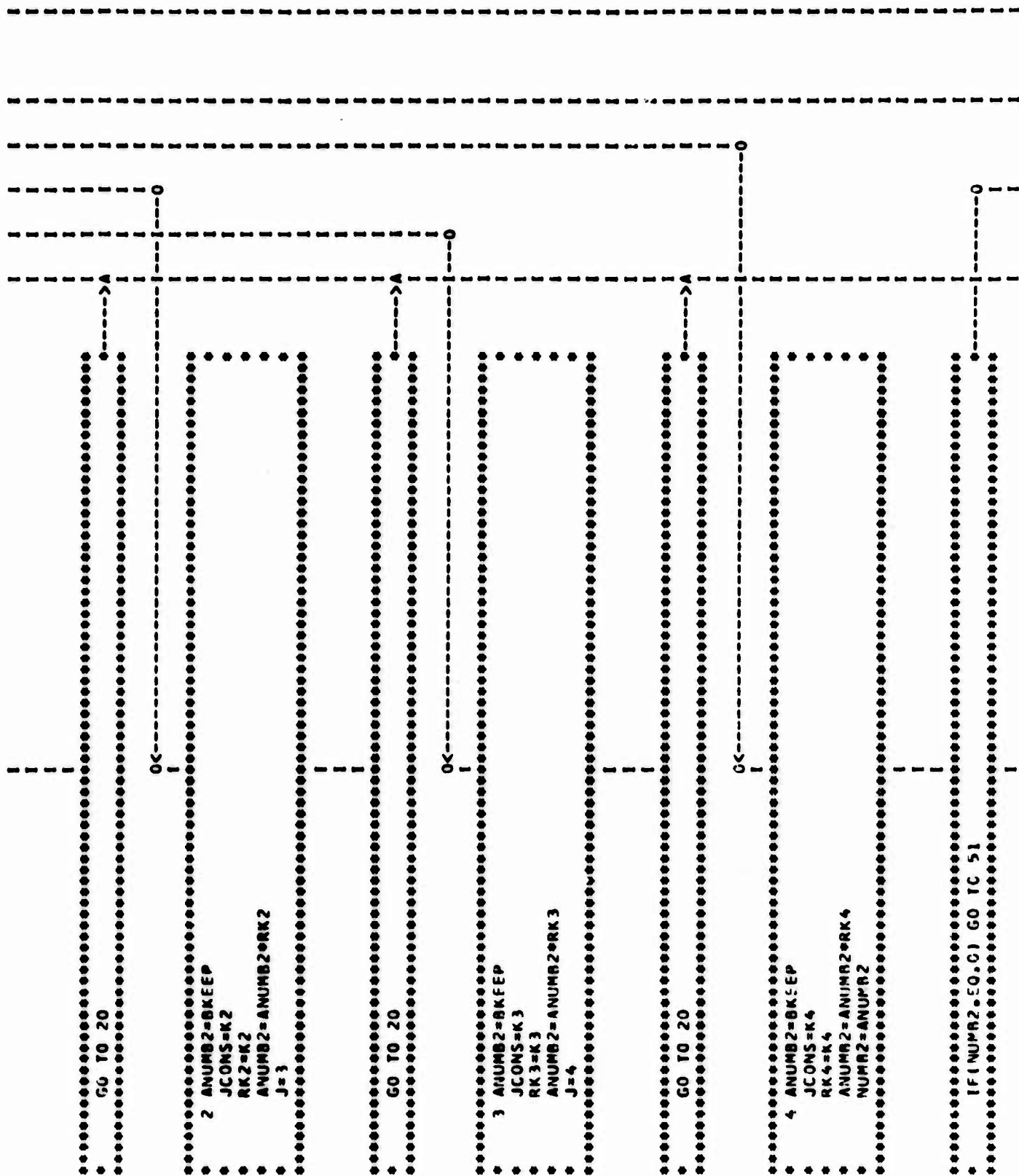
  IF(NUMB2.EQ.10)GO TO 11

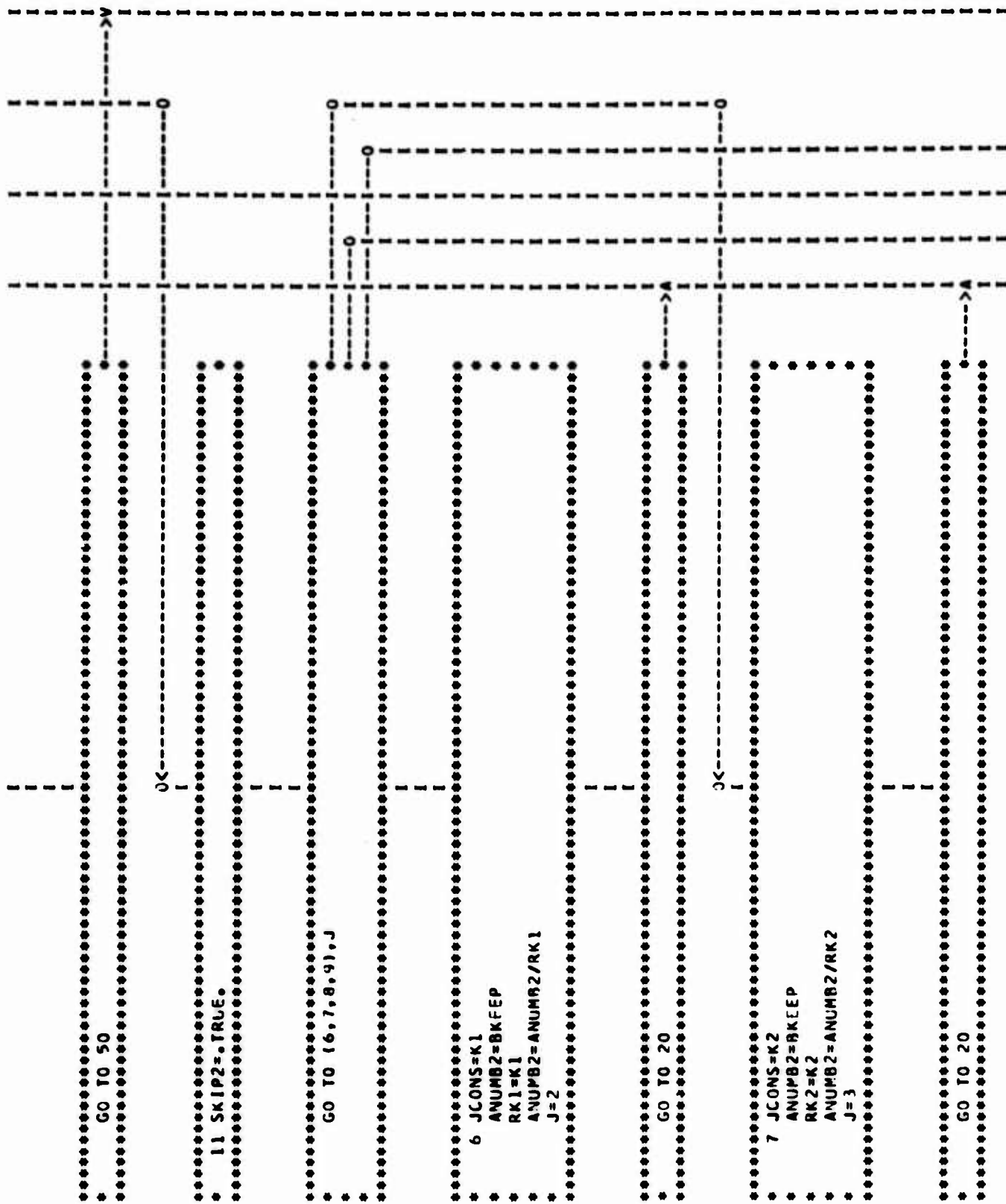
  GO TO 50

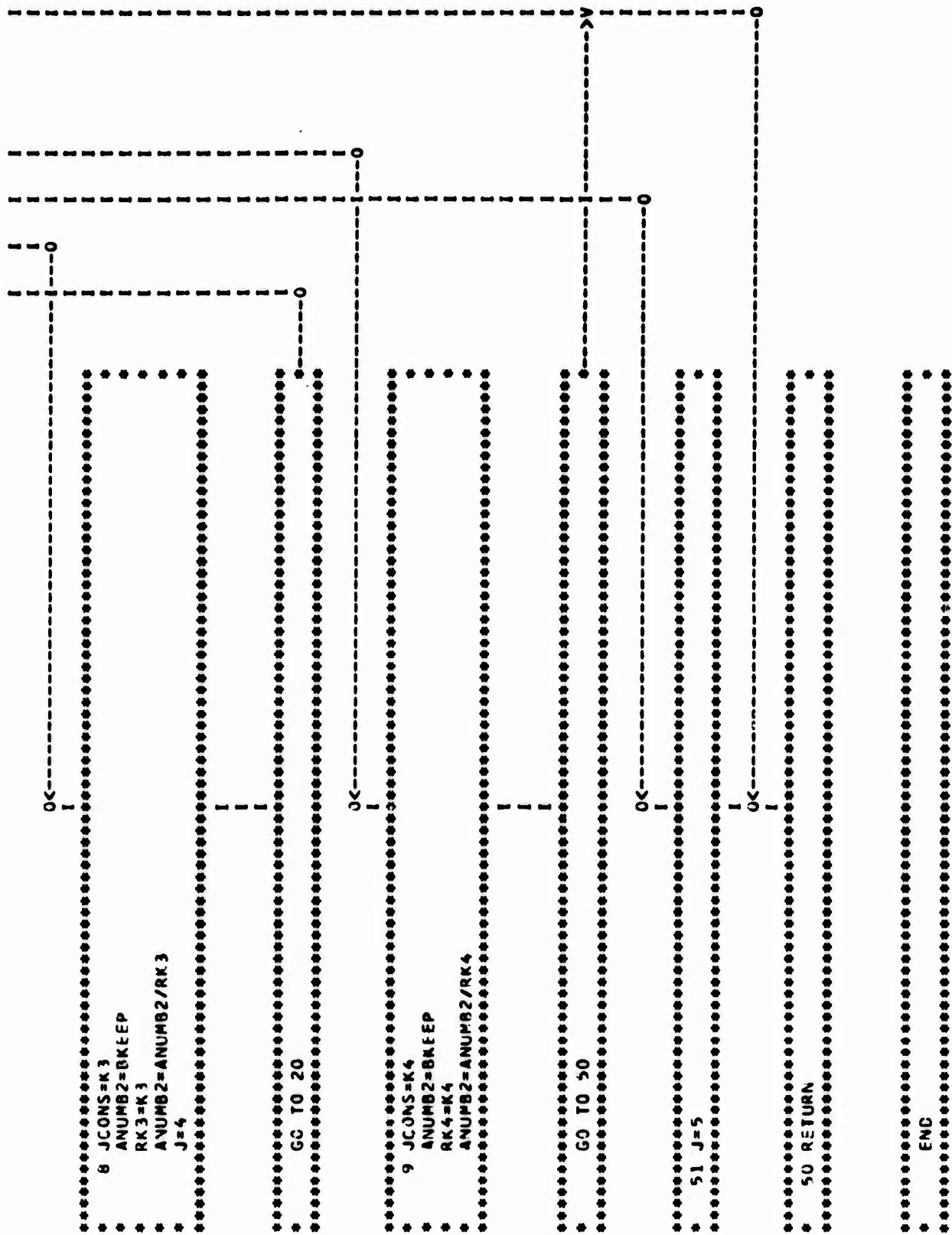
  10 GOTC(1,2,1,4),J

  1 ANUMB2=BKEEP
  JCONS=K1
  RK1=K1
  ANUMB2=ANUMB2+RK1
  J=2

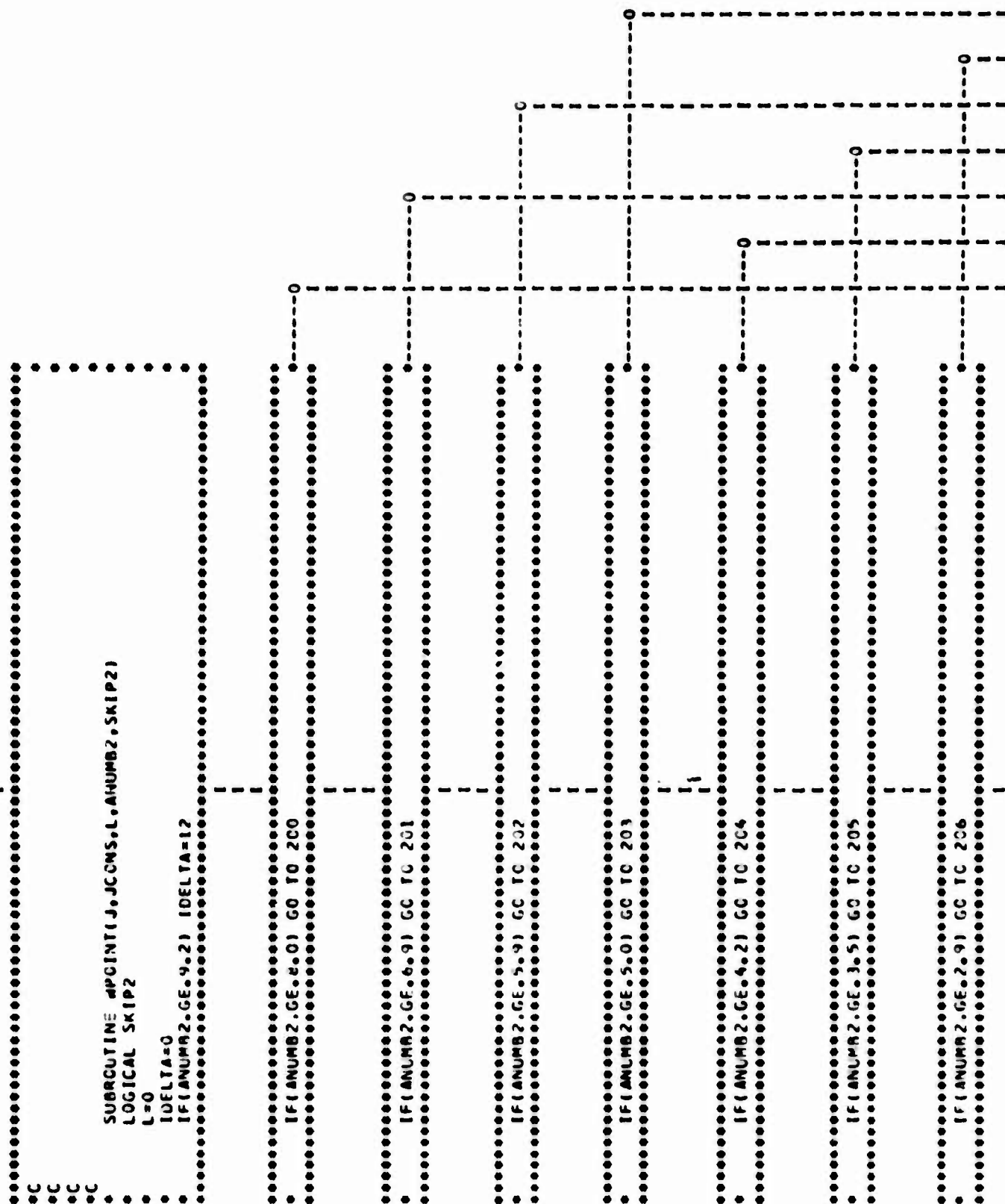
```







(ENTRANCE)



```

*****
* IF (ANUMB2,GE,2.4) GO TO 207
*****

*****
* IF (ANUMB2,GE,1.9) GO TO 208
*****

*****
* IF (ANUMB2,GE,1.5) GO TO 209
*****

*****
* IF (ANUMB2,GE,1.2) GO TO 210
*****

*****
* GO TO 211
*****

*****
* 200 IF (ANUMB2,LT,9.2) IDELTA=11
*****

*****
* 201 IF (ANUMB2,LT,8.0) IDELTA=10
*****

*****
* 202 IF (ANUMB2,LT,6.9) IDELTA=9
*****

*****
* 203 IF (ANUMB2,LT,5.9) IDELTA=8
*****

```



```

*****
* IF(SKIP2) GO TO 41
*****

*****
* IF(JCCNS.EQ.1)GOTO 1
*****

*****
* IF(JCCNS.EQ.10)GOTO 10
*****

*****
* IF(JCCNS.EQ.100)GOTO 100
*****

*****
* IF(JCCNS.EQ.1000)GOTO 1000
*****

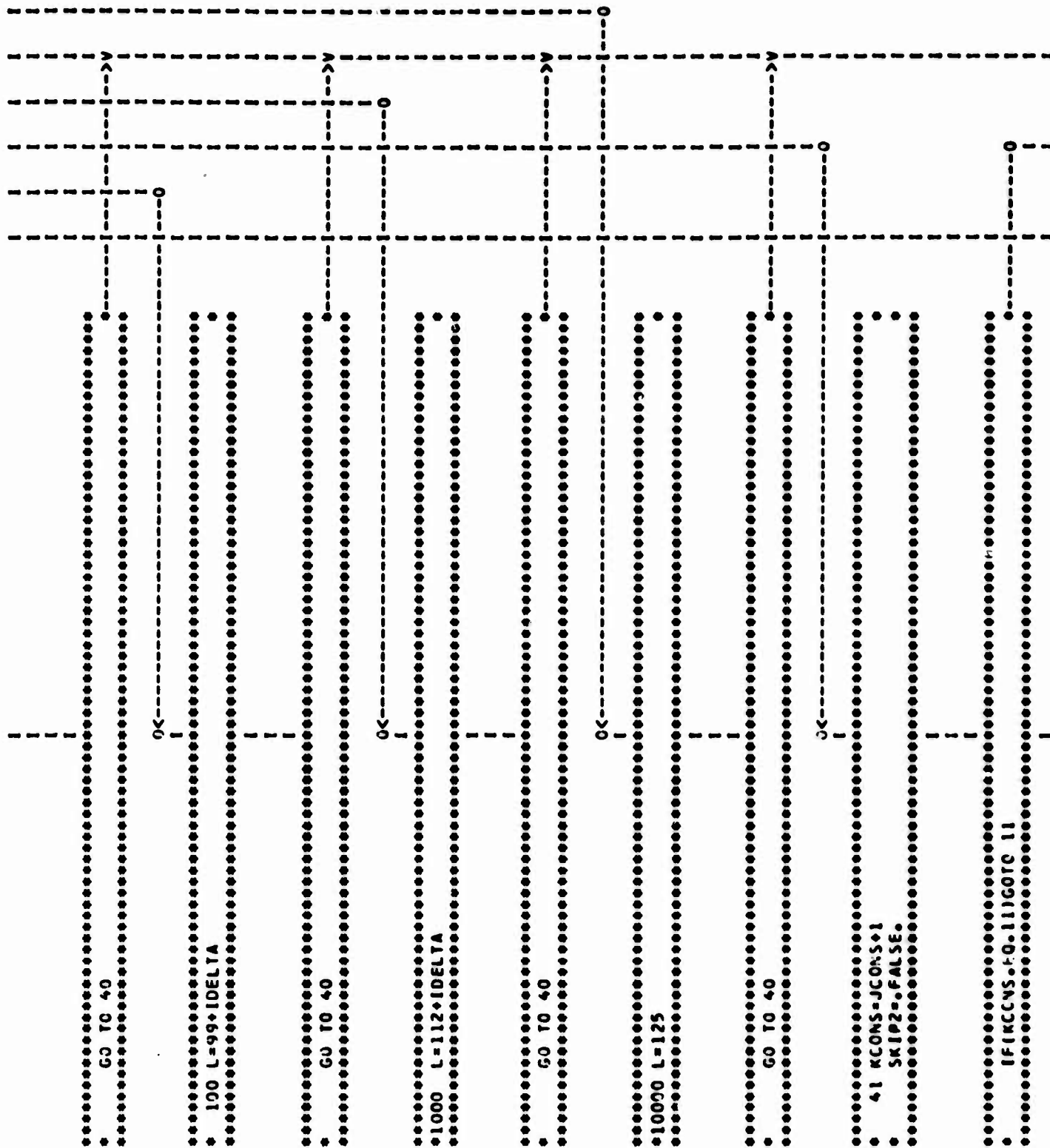
*****
* IF(JCCNS.EQ.10000)GOTO 10000
*****

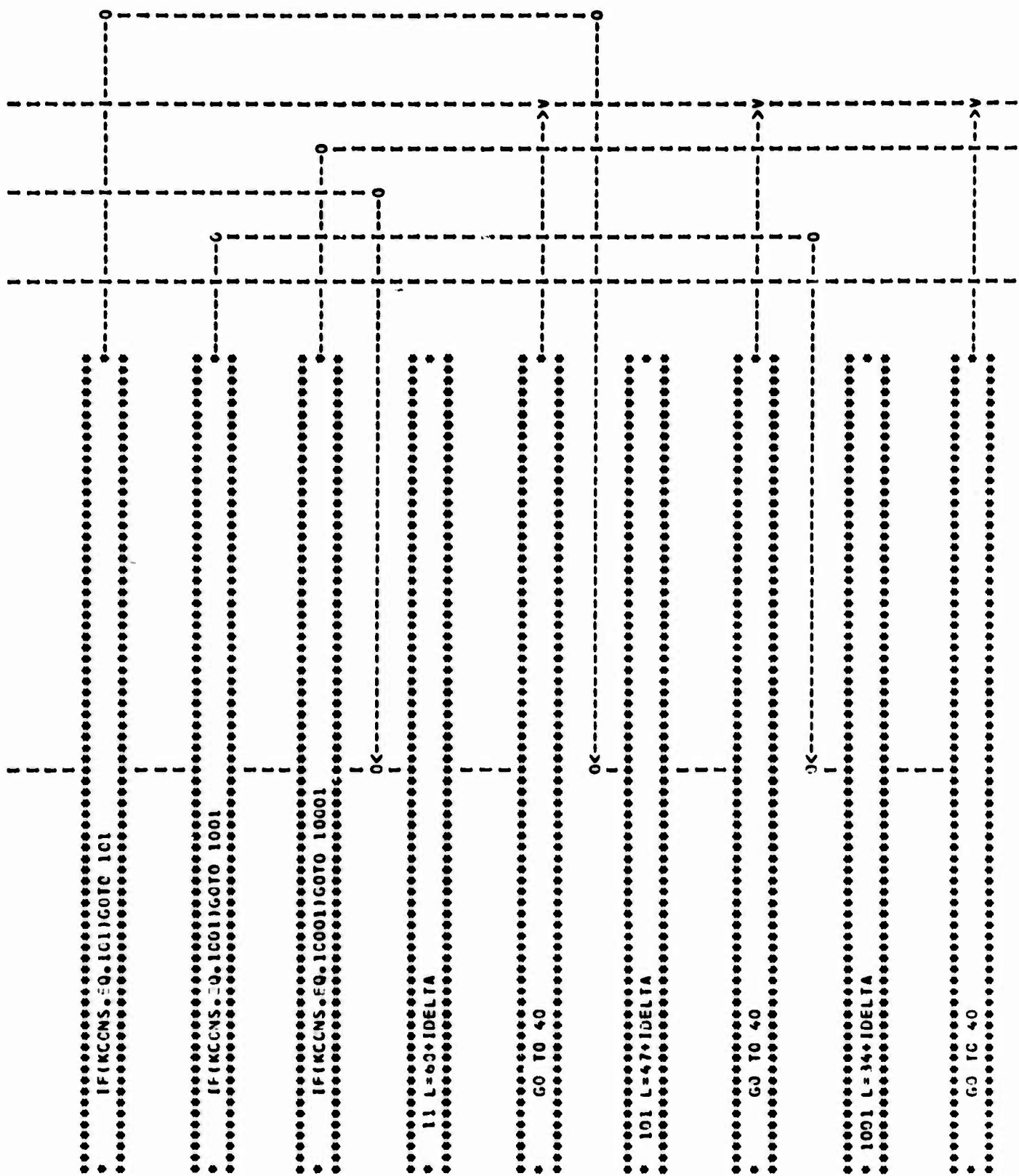
*****
*1 L=73+IDELTA
*****

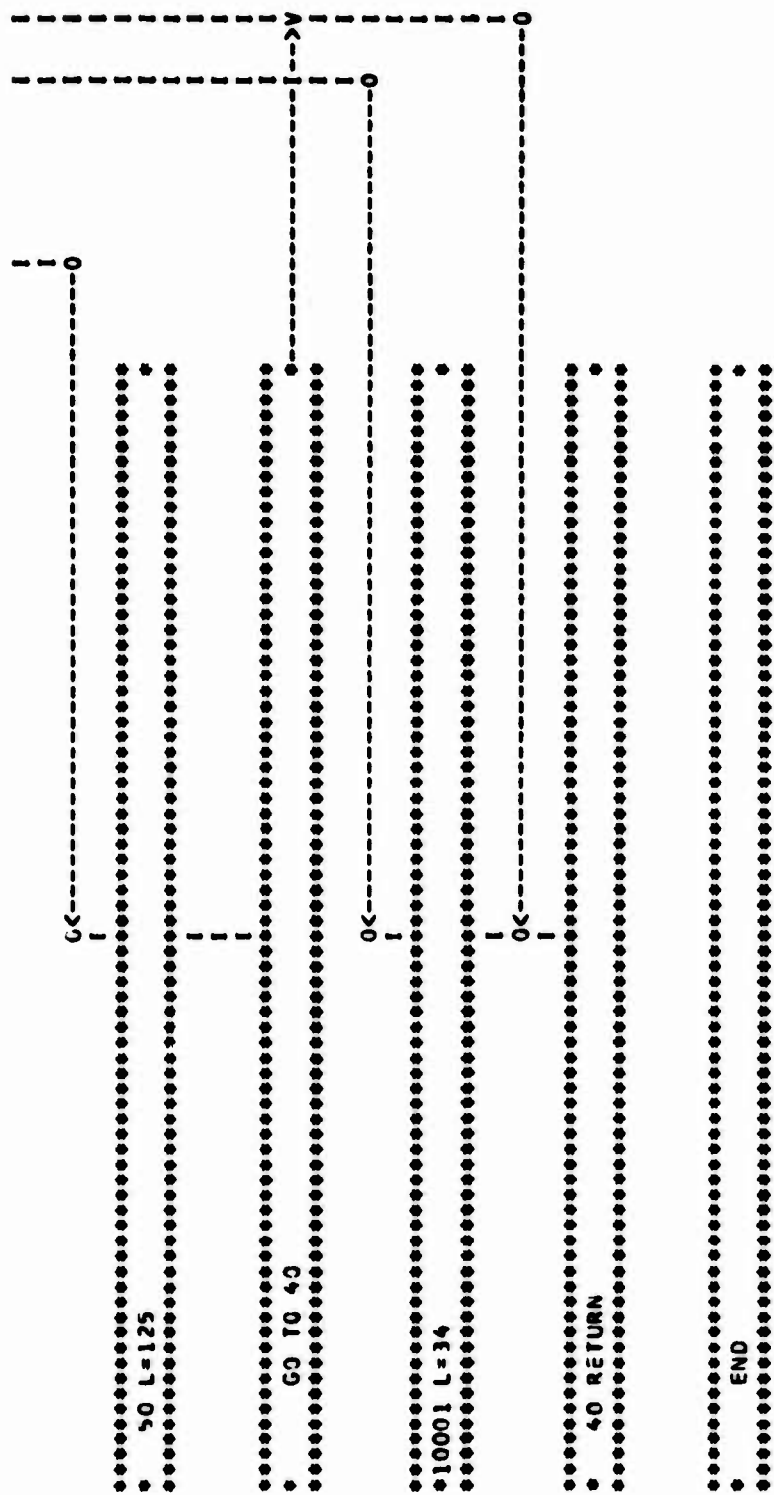
*****
* GO TO 40
*****

*****
*10 L=H6+IDELTA
*****

```







(ENTRANCE)

```

*****
*C
*C
*C
*C
* SUBROUTINE SPCINT(I,ANUMB1,LESS,ICONS,L,POINT,SKIPI,NO,XA,XB,STAR)
* DIMENSION XA(200),XR(200)
* INTEGER POINT(130,100),DASH,BLANK,STAR,C
* INTEGER STAR1,STAR2
* DATA STAR1/1MC/,STAR2/1H*/
* DATA BLANK/1H /,DASH/1H-/,
* LOGICAL SKIPI
* AKEEP=ANUMB1
* ANUMB1=ABS(AKEEP)
* KOOL=C
*****

```

```

*****
* IF I.EQ.51G0 TO 100C
*****

```

```

*****
* IF (ANUMB1.GE.7.1)KOOL=7
*****

```

```

*****
* IF (ANUMB1.GE.5.01G0 TO 1011
*****

```

```

*****
* IF (ANUMB1.GE.4.01G0 TO 1111
*****

```

```

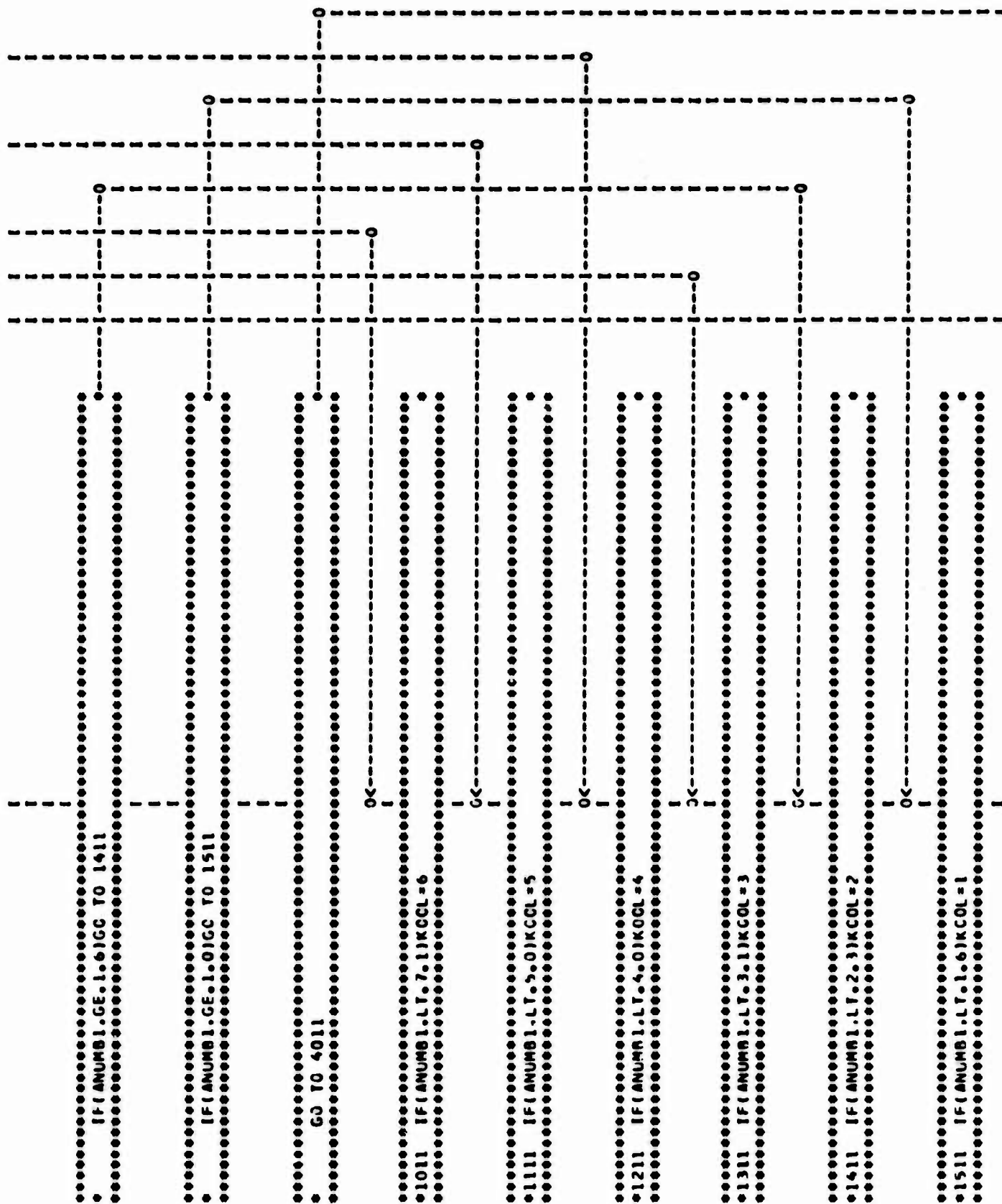
*****
* IF (ANUMB1.GE.3.11G0 TO 1211
*****

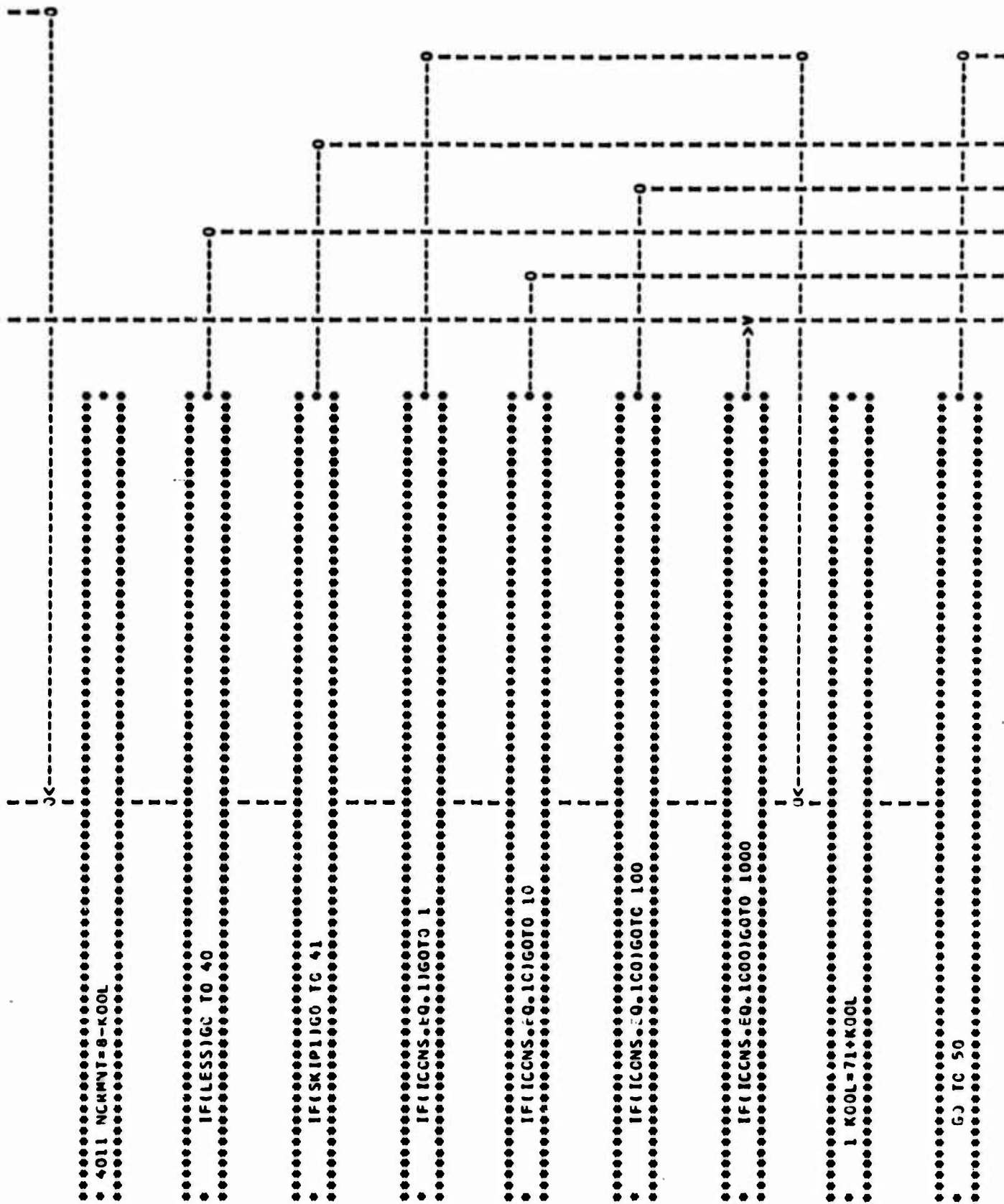
```

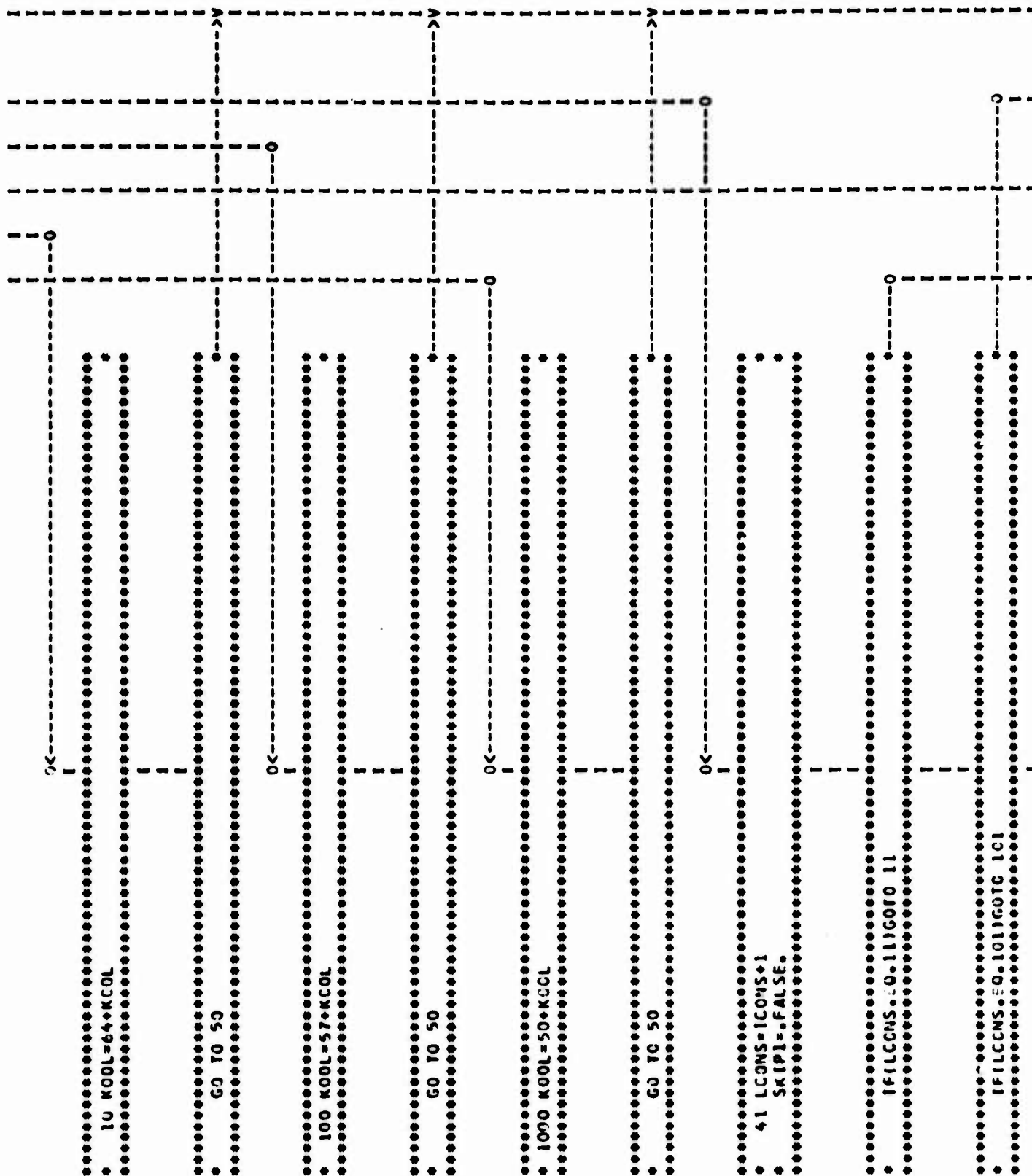
```

*****
* IF (ANUMB1.GE.2.31G0 TO 1311
*****

```



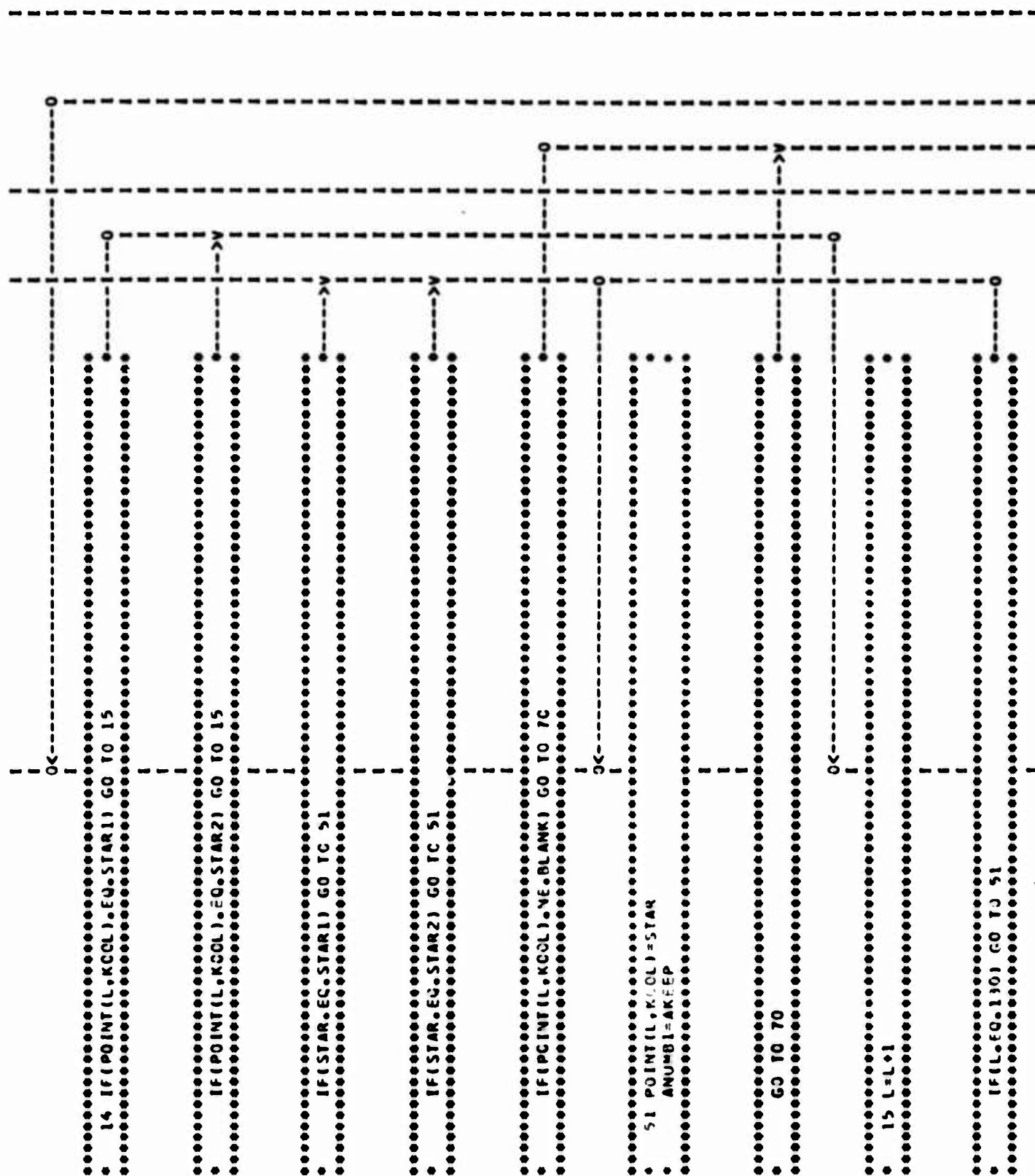


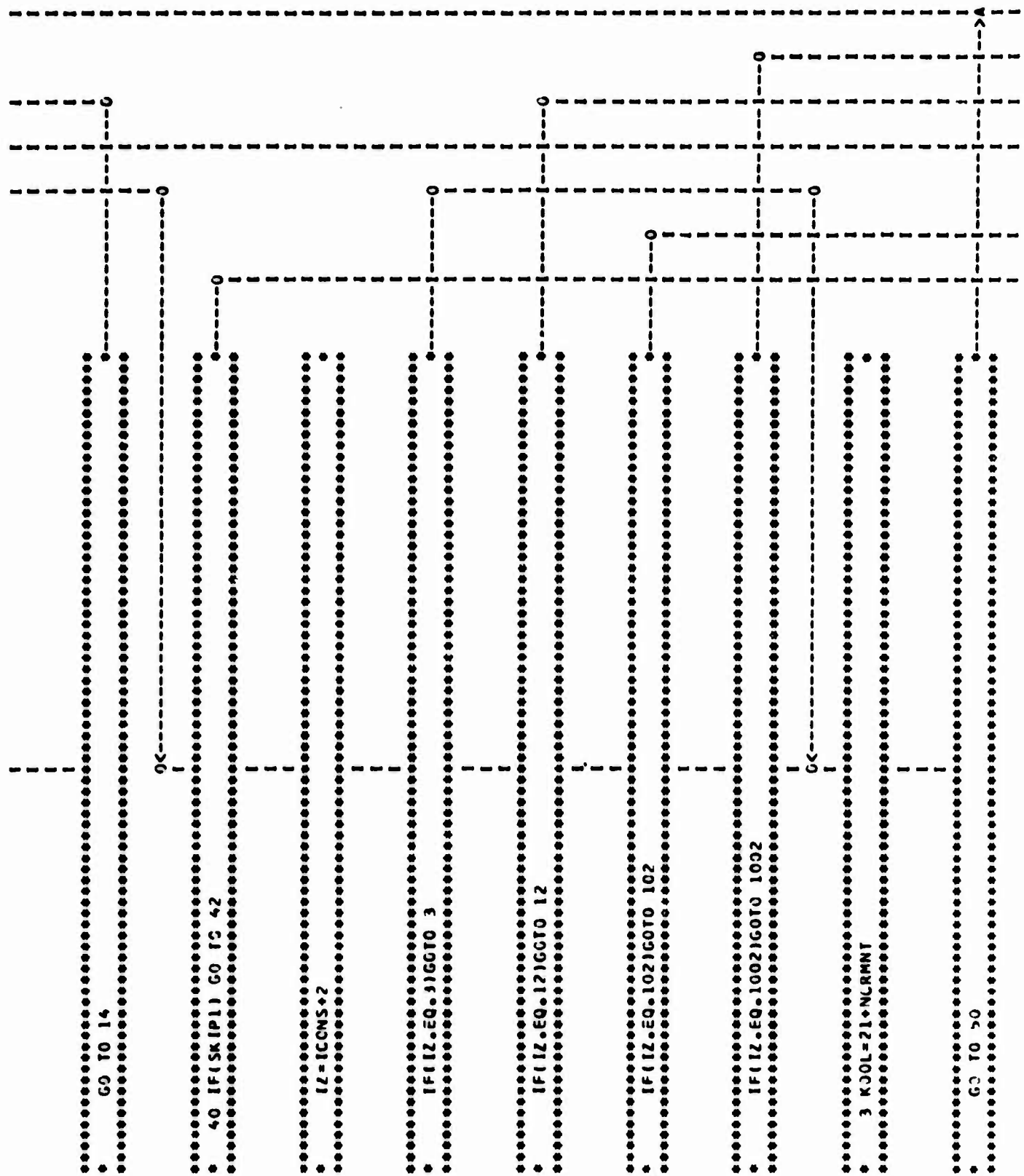


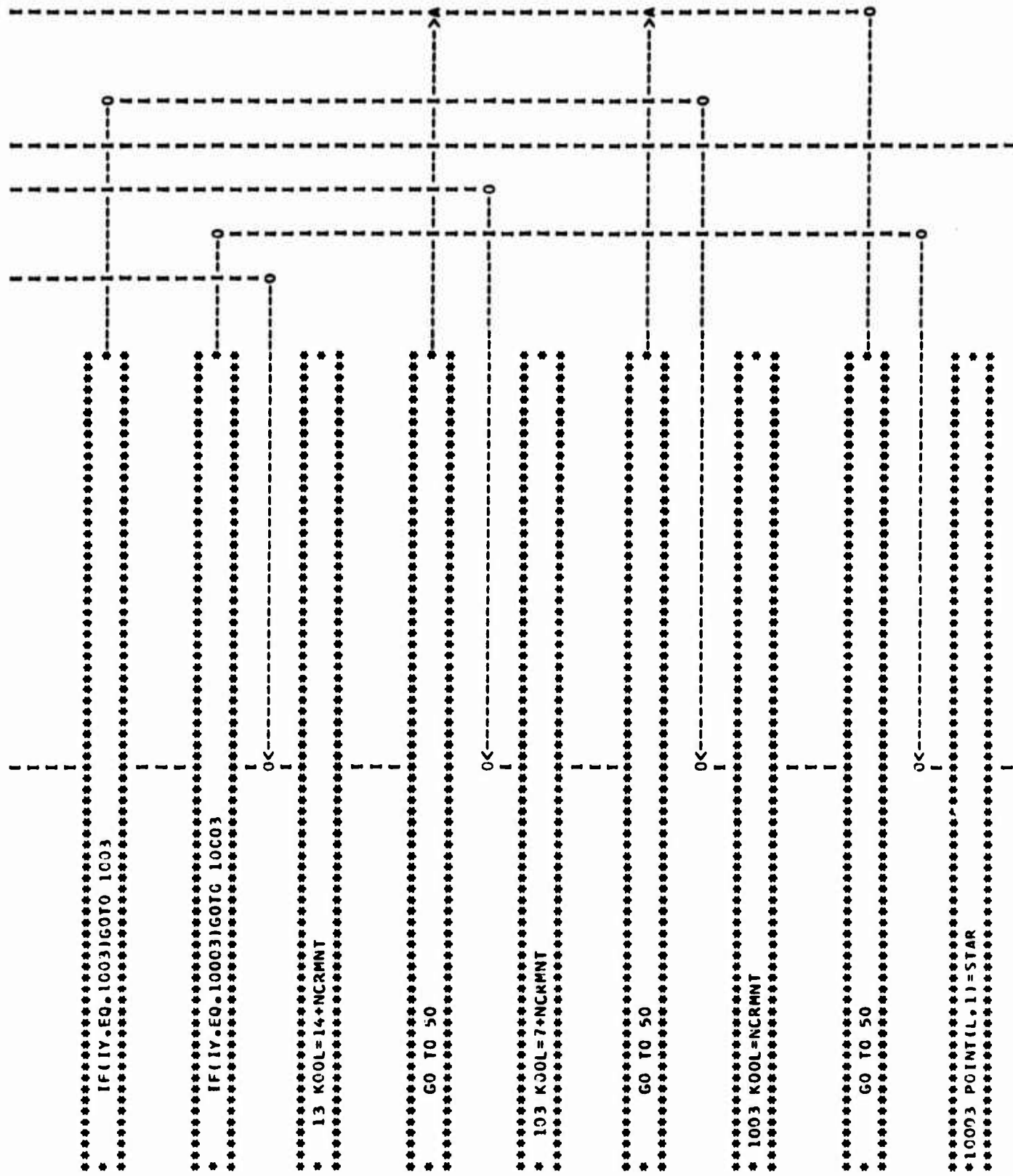
```

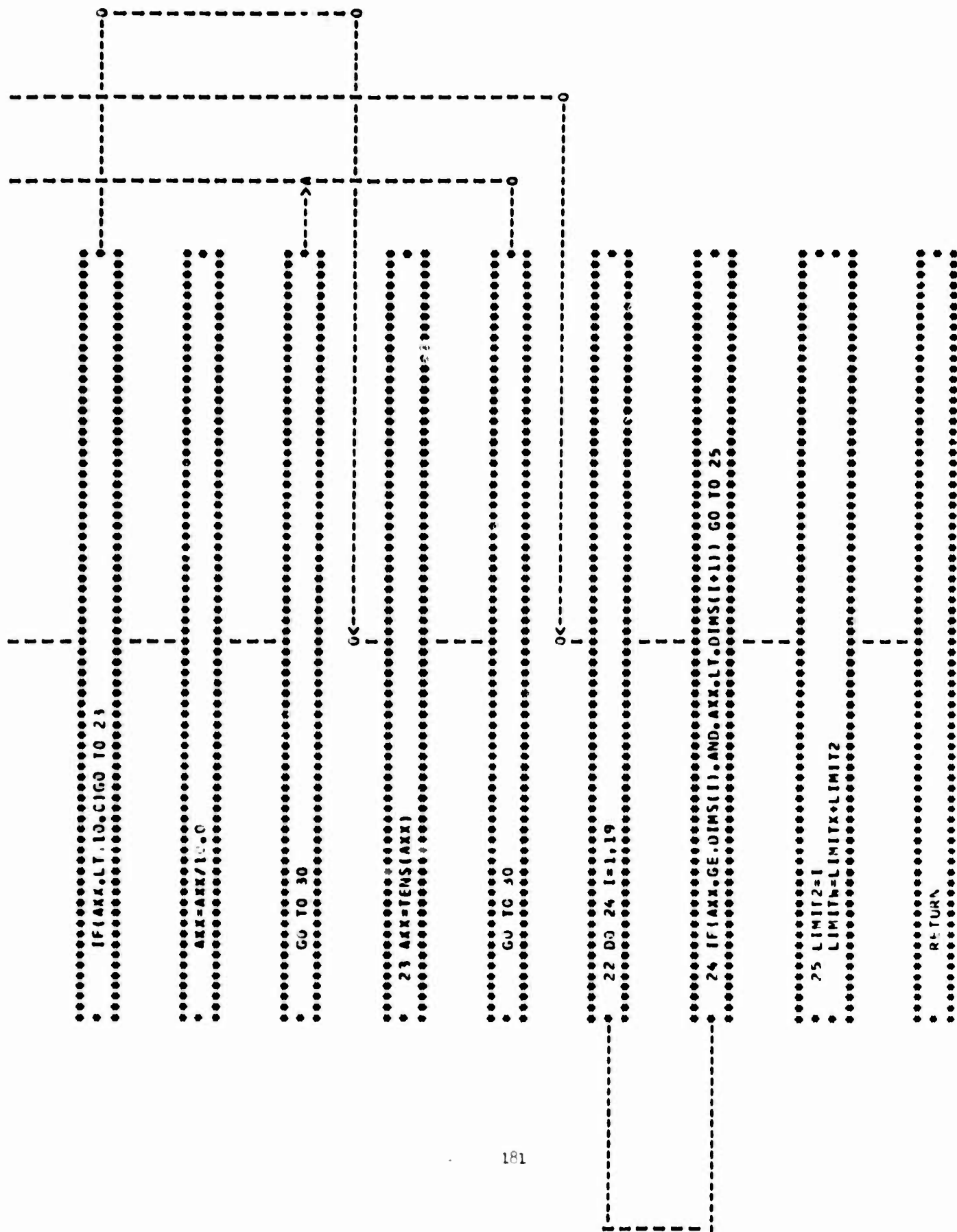
.....
* IF(LCCMS.EQ.1001)GOTO 1001
.....
0<-----0
.....
* 11 KCOL=78+KCOL
.....
* GO TO 50
.....
0<-----0
.....
* 101 KCOL=85+KCOL
.....
* GO TO 50
.....
0<-----0
.....
* 1001 KCOL=92+KCOL
.....
0<-----0
.....
* 50 LESS=.FALSE.
.....
* IF(PCINT(L,KCOL).EQ.DASH) GO TO 51
.....
* IF(PCINT(L,KCOL).EQ.0 ) GO TO 51
.....

```









END

(ENTRANCE)

```
.....
* C
* C
* SUBROUTINE BLANKR (POINT,N1,N2,LIM1)
* INTEGER POINT(N1,N2),BLANK,DASH,Q
* COMMON /INFO4/ Q,BLANK,DASH
* DATA BLANK/1H /,DASH/1H-/,Q/1H1/
* .....
```

```
.....
DO 51 M=1,N1
.....
```

```
.....
DO 51 N=1,N2
.....
```

```
.....
51 POINT(M,N)=BLANK
.....
```

```
.....
DO 52 M=1,N1
.....
```

```
.....
POINT(M,1)=DASH
.....
```

```
.....
52 POINT(M,N2)=DASH
.....
```

```
.....
DO 53 M=1,N2
.....
```


(ENTRANCE)

```

*****
**C
**C
**  SURROUTINE NEWPLT(X,Y,Z,LET)
**  INTEGER POINT(130,100),POINTF(120,58),STAR,Q
**  DIMENSION ALIM(7)
**  DIMENSION Y(3000),Z(3000)
**  DIMENSION SAVE1(100,100),SAVE2(100,100)
**  COMMON /INFO4/ Q,BLANK,DASH
**  COMMON /INFO2/ALIM
**  COMMON SAVE1,SAVE2,POINT
**  EQUIVALENCE(POINT(1,1),POINTF(1,1))
**  DATA STAR/1H*/Q/1H/
**  1 FORMAT(1H1)
**  CALL BLANKR(POINTF,120,58,120)
*****

```

```

*****
**  DO 10 I=1,3000
*****

```

```

*****
**  IF(Y(I).EQ.O.C) GO TO 11
*****

```

```

*****
**  GO TO 12
*****

```

```

*****
**  11 IF(Z(I).EQ.O.C) GO TO 10
*****

```

```

*****
**  12 CALL LOGGR(Y(I),LIMITW)
*****

```

```

*****
**  IF(LIMITW.EQ.Q) GO TO 10
*****

```

```

      CALL LINEAR(X,LIMITY,Z(I))
      IF(LIMITY.EQ.0) GO TO 10
      POINTF(LIMITY,LIMITY)=STAR
      10 CONTINUE
      DO 40 I=20,100,20
      DO 40 J=1,58
      40 POINTF(I,J)=Q
      WRITE(6,1)
      30 FORMAT(1X,6(IPIF10.1,1X,3MR/S.6X),1PIE1C.1)
      J=X
      WRITE(6,51)J,LET,(POINTF(I,1),I=1,120)
      51 FORMAT(1X,14, A4,1X,12CA1)
      WRITE(6,20)((POINTF(I,J),I=1,120),J=2,57)
      J=-X
      WRITE(6,51)J,LET,(POINTF(I,58),I=1,120)
      20 FORMAT(10X,12CA1)
      WRITE(6,30)JALIM

```

1
1
1

.....
* R F T U R N *
.....

.....
* E N D *
.....

(ENTRANCE)

```

.....
* C .....
* C .....
* SUMMULTINE LINEAR(X,LIMIT,Y)
* DIMENSION ENCRMT(59)
* LIMITY=0
* FLIMIT=X*2.
* DELTA=FLIMIT/58.
* ENCRMT(1)=X
.....

```

```

.....
DO 10 I=2,58
.....

```

```

.....
10 ENCRMT(I)=ENCRMT(I-1)-DELTA
.....

```

```

.....
ENCRMT(59)=-X
.....

```

```

.....
DO 11 I=1,58
.....

```

```

.....
11 IF(Y.LE.ENCRMT(I).AND.Y.GE.ENCRMT(I+1))GO TO 20
.....

```

```

.....
GO TO 40
.....

```

```

.....
20 LIMITY=I
.....

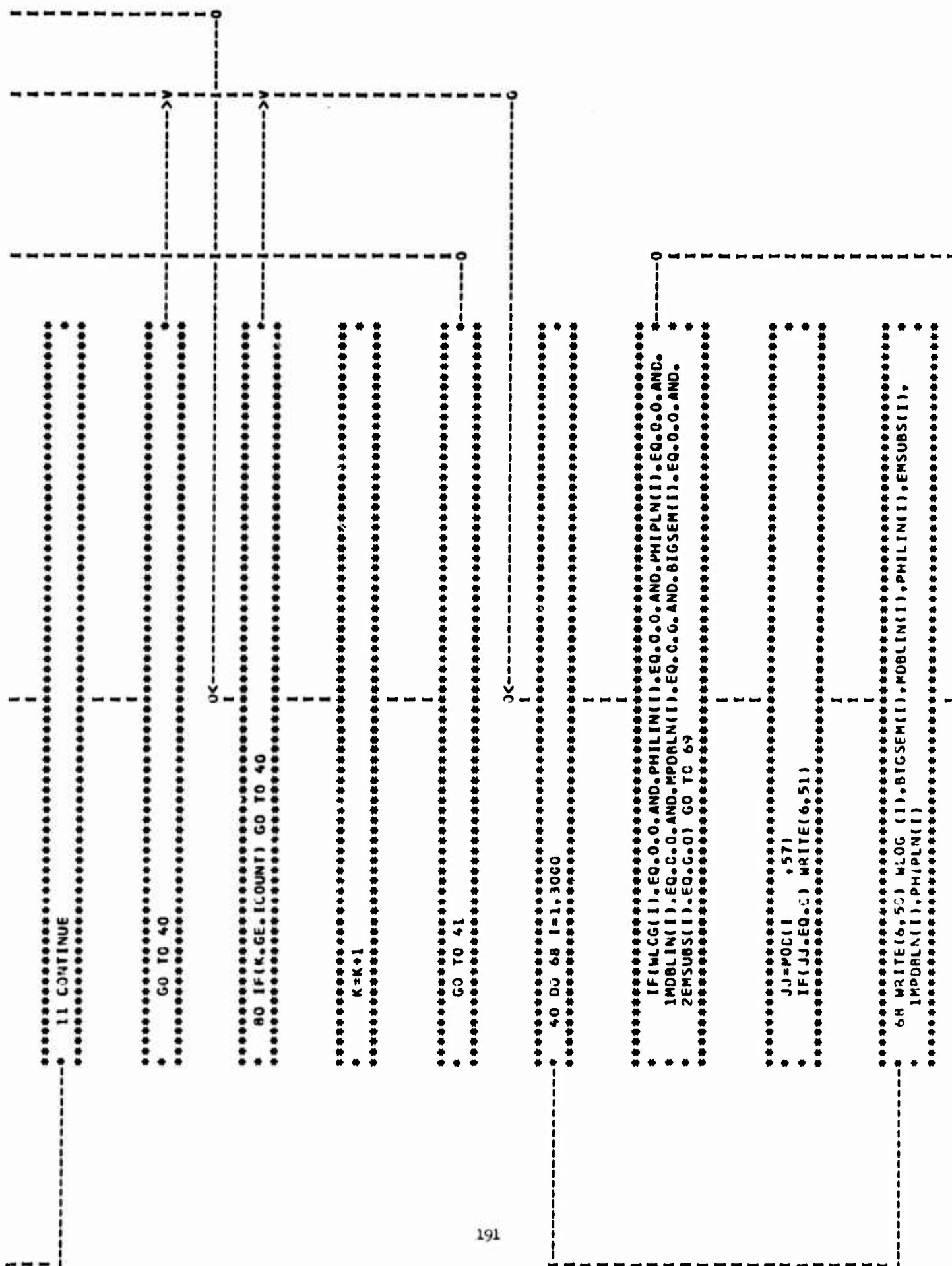
```



```

(ENTRANCE)
I
I
*****
*C
*C
* SURROUTINE SWEEP(XOMEGA)
* DIMENSION OMEGA(17),SWEPER(17)
* COMMON/INFO3/OMEGA
* DATA (SWEPER(I),I=1,17) /0.01,0.05,0.1,0.2,0.3,0.4,0.5,0.6,0.7,
* 0.8,0.9,1.0,2.0,4.0,6.0,8.0,10.0/
*****
I
I
I
I
I
*****
DO 70 K=1,17
*****
*****
70 OMEGA(K) = SWEPER(K)*XOMEGA
*****
I
I
I
*****
* RETURN
*****
*****
* END
*****

```



```

*****
* 69 WRITE(6,14)
*      CALL NEWPLT(DEGREE,WLOG,PHILIN,LET2 )
*****

*****
*      IF(IEXP(1).EQ.0) GO TO 30
*****

*****
*      JJ=IEXP(1)
*****

*****
*      DO 34 I=1,JJ
*****

*****
*      34 CALL EXPLOT(WLOG,3000,PHILIN,3000,DEGREE,LET2)
*****

*****
*      30 CALL NEWPLT(DBLIN,WLOG,MDBLIN,LET1)
*****

*****
*      IF(IEXP(2).EQ.0) GO TO 31
*****

*****
*      JJ=IEXP(2)
*****

*****
*      DO 35 I=1,JJ
*****

```

```

.....
15 CALL EXPLUT(WLOG,3000,MDRLIN,3000,ORLIN,LET1)
.....
OK-----0
.....
31 WRITE(6,15)
CALL NEMPT(DEGREE,WLOG,PHIPLN,LET2)
.....
.....
IF(IEXP(3).EQ.0) GO TO 32
.....
JJ=IEXP(3)
.....
DO 36 I=1,JJ
.....
16 CALL EXPLUT(WLOG,3000,PHIPLN,3000,DEGREE,LET2)
.....
OK-----0
.....
32 CALL NEMPT(ORLIN,WLOG,MPDBLN,LET1)
JJ=IEXP(4)
.....
DO 37 I=1,JJ
.....

```

```

      17 CALL EXPLGT(MLOG,1000,MPDULN,3000,CBLIM,LE11)
      IF(IEXP(4).EQ.0) GO TO 33
      51 FORMAT(1H1,5X,SHOMEGA,13X,3H/M/,16X,6H/M/-DB,10X,3HPHI,20X,4H/M/P,
      10X,7H/M/P-DB,14X,5HPHI-P,/)
      50 FORMAT(2X,1PIE12.5,5X,1PIE12.5,6X,1PIE12.5,5X,1PIE12.5,11X,
      1PIE12.5,3X,1PIE12.5,9X,1PIE12.5)
      14 FORMAT(1H1,//////////,25X,60HTHE FOLLOWING PLOTS ARE NOR
      15 FORMAT(1H1,//////////,25X,36HTHE FOLLOWING PLOTS ARE POP
      16V DATA
      33 RETURN
      END

```

(ENTRANCE)

```

*****
C
C
SUBROUTINE FIGUR (KONT,OMEGA,KOUNT)
DIMENSION SAVE(100,100),SAVE2(100,100)
DIMENSION ACOEFF(100),RCOEFF(100),AAAAA(99),BBBBB(99)
DIMENSION WLOG(300),PHILIN(300),MDRLN(300),PHIPLN(300),
*MPDBLN(300),BIGSEM(300),EMSUBS(300),OMEGA(KONT)
REAL MDRLN,MPDBLN
INTEGER P,INT(130,100)
COMMON/FREK/ACOEFF,RCOEFF
EQUIVALENCE(AAAAA,1),ACOEFF(2),(BBBBB,1),BCOEFF(2)
COMMON/INFO6/EMSUBS
COMMON SAVE1,SAVE2,POINT
EQUIVALENCE (WLOG,1),SAVE1(1),(PHILIN,1),SAVE2(1),
1 (SAVE1(300),MDRLN(1)),(SAVE2(300),PHIPLN(1)),
2 (SAVE1(600),MPDBLN(1)),(SAVE2(600),BIGSEM(1))
*****

```

```

*****
DO 21 I=1,KONT
*****

```

```

*****
CALL FIGURE(OMEGA(1),ACOEFF(1),AAAAA,EMOFAS,PHIAS)
CALL FIGURE(OMEGA(1),RCOEFF(1),BBBBB,EMOFBS,PHIBS)
BIGEM=ABS(EMOFAS/EMOFBS)
WLOG(KONT)=BIGEM
PHIENG=PHIAS-PHIBS
IF(PHIEND.LT.-180.) PHIEND=PHIEND+360.
IF(PHIEND.GT.180.) PHIEND=PHIEND-360.
PHILIN(KONT)=PHIEND
PHIENG = PHIEND * 0.0174533
REEL = BIGEM * COS ( PHIEND )
EIMAG= BIGEM * SIN ( PHIEND )
EMSUBP = SQRT ( ( REEL**2 ) + ( OMEGA ( 1 ) * EIMAG ) **2 ) )
EMSUBS(KONT)=EMSUBP
QUANZ=OMEGA(1) * EIMAG
PHIPEE=ATAN2(QUANZ,REEL)
PHIPEE = PHIPEE * 57.2957795131
IF(PHIPEE.LT.-180.) PHIPEE = PHIPEE + 360.
IF(PHIPEE.LT.-180.) PHIPEE=PHIPEE+360.
IF(PHIPEE.GT.180.) PHIPEE=PHIPEE-360.
PHIPLN(KONT)=PHIPEE
DEEBE=20.0*(ALOG10(BIGEM))
MDRLN(KONT)=DEEBE
DEEBEP=20.0*(ALOG10(EMSUBP))
MPDBLN(KONT)=DEEBEP
KOUNT=KOUNT+1
*****

```

```

-----
21 CONTINUE
-----
RETURN
-----
END
-----

```

(ENTRANCE)

```

C
C
C SUBROUTINE FIGURE(X,Y,Z,EMCFAS,PHIAS)
C DIMENSION Z(99)
C REELAS=Y
C I=2
C A=-1.0

```

```

C
C
C 21 REELAS=(1211)*(X+111)*A)*REELAS
C

```

```

C
C
C 25 I=1+2
C

```

```

C
C
C IF(1.GT.99) GO TO 22
C

```

```

C
C
C A=-A
C

```

```

C
C
C IF(211.EQ.0.)GO TO 25
C

```

```

C
C
C GO TO 21
C

```

```

C
C
C 27 I=1
C A=1.0
C ALMAGA=0.0

```



```

0<-----
.....
•   24 AIMAGA=(Z(1)*(X**I)+A)*AIMAGA
.....
0<-----
.....
•   26 I=I+2
.....
.....
.....
IF(I.GT.94) GO TO 23
.....
.....
.....
      A=-A
.....
.....
IF(Z(1).EC.O.)GO TO 26
.....
.....
.....
      GO TO 24
.....
0<-----
.....
.....
21 EMOFAS=SQRT((REELAS**2)+(AIMAGA**2))
• PHIAS=ATAN2(AIMAGA,REELAS)
• PHIAS = PI-IAS*57.2957795131
• IF(PHIAS.LT.O.) PHIAS=PHIAS+360.
.....
.....
RETURN
.....

```

• **QW3** •

(ENTRANCE)

```
.....  
C  
C  
.....  
SURROUTINE LGGER(XX,LIMITM)  
.....  
DIMENSION ALIM(7),DIMS(20)  
.....  
EQUIVALENCE(WASTE,AXX)  
.....  
COMMON/INFO2/ALIP  
.....  
TENSIX)=X*10.C  
.....  
DATA (DIMS(1),1-1,20) /1.C,1.15,1.3,1.5,1.7,1.95,2.25,2.5,2.85,  
.....  
3.15,3.5,3.95,4.5,4.85,5.4,6.05,6.8,7.7,8.8,9.99999/  
.....  
LIMITM=0  
.....  
LIMITX=0  
.....  
LIMIT2=0  
.....
```

DO 20 I=1,6

AA=2*(I-1)

20 IF(XX.LT.ALIM(I+1).AND.XX.GE.ALIM(I)) GO TO 21

RETURN

21 LIMITX=TENSIX

WASTE=ABS(XX)

IF(WASTE-10.0.0) RETURN

30 IF(AXX.GE.1.C.AND.AXX.LT.10.0) GO TO 22

(ENTRANCE)

```

.....
* C
* C
* C
* C
* SURROUTINE PREPAR(POINT)
* INTEGER P(130,100),PRT
* DATA PRT /6/
* WRITE(6,1)
* 1 FORMAT(1H1)
* WRITE(6,11)
* 11 FORMAT(1//)
* WRITE(PRT,14)
* 14 FORMAT(75X,8HLOG PLOT,75X,42HCOMPLEX FREQUENCY PLANE,LEFT HAND QU
* ADRANT,75X,9H(RAD/SEC) )
* WRITE(6,12)
* 12 FORMAT(32X,5H10000,9X,4H1000,9X,3H100,1CX,2H10,12X,1H1,11X,2M.1,10
* X,14H.01C---J-OMEGA,7 18X,11HMINUS SIGMA)
* 13 FORMAT(32X,5H10000,9X,4H1000,9X,3H100,1CX,2H10,12X,1H1,11X,2M.1,10
* X,14H.01C---J-OMEGA,7 18X,11H PLUS SIGMA)
.....

```

```

.....
* C
* C
* C
* C
* DO 50 I=1,130
* .....
* 50 POINT(I,50)=POINT(I,51)
* .....

```

201

```

.....
* WRITE(PRT,10)POINT
* 10 FORMAT(1X,130A1)
* WRITE(6,11)
* WRITE(PRT,15)
* 15 FORMAT(75X,8HLOG PLOT,75X,42HCOMPLEX FREQUENCY PLANE,RIGHT HAND QU
* ADRANT,75X,9H(RAD/SEC) )
.....

```

```

.....
* RETURN
* .....

```

0 0.5 1 1.5 2 2.5 3 3.5 4 4.5 5 5.5 6 6.5 7 7.5 8 8.5 9 9.5 10 10.5 11 11.5 12 12.5 13 13.5 14 14.5 15 15.5 16 16.5 17 17.5 18 18.5 19 19.5 20 20.5 21 21.5 22 22.5 23 23.5 24 24.5 25 25.5 26 26.5 27 27.5 28 28.5 29 29.5 30 30.5 31 31.5 32 32.5 33 33.5 34 34.5 35 35.5 36 36.5 37 37.5 38 38.5 39 40 40.5 41 41.5 42 42.5 43 43.5 44 44.5 45 45.5 46 46.5 47 47.5 48 48.5 49 50 50.5 51 51.5 52 52.5 53 53.5 54 54.5 55 55.5 56 56.5 57 57.5 58 58.5 59 60 60.5 61 61.5 62 62.5 63 63.5 64 64.5 65 65.5 66 66.5 67 67.5 68 68.5 69 70 70.5 71 71.5 72 72.5 73 73.5 74 74.5 75 75.5 76 76.5 77 77.5 78 78.5 79 80 80.5 81 81.5 82 82.5 83 83.5 84 84.5 85 85.5 86 86.5 87 87.5 88 88.5 89 90 90.5 91 91.5 92 92.5 93 93.5 94 94.5 95 95.5 96 96.5 97 97.5 98 98.5 99 100 100.5 101 101.5 102 102.5 103 103.5 104 104.5 105 105.5 106 106.5 107 107.5 108 108.5 109 109.5 110 110.5 111 111.5 112 112.5 113 113.5 114 114.5 115 115.5 116 116.5 117 117.5 118 118.5 119 119.5 120 120.5 121 121.5 122 122.5 123 123.5 124 124.5 125 125.5 126 126.5 127 127.5 128 128.5 129 129.5 130 130.5 131 131.5 132 132.5 133 133.5 134 134.5 135 135.5 136 136.5 137 137.5 138 138.5 139 139.5 140 140.5 141 141.5 142 142.5 143 143.5 144 144.5 145 145.5 146 146.5 147 147.5 148 148.5 149 149.5 150 150.5 151 151.5 152 152.5 153 153.5 154 154.5 155 155.5 156 156.5 157 157.5 158 158.5 159 159.5 160 160.5 161 161.5 162 162.5 163 163.5 164 164.5 165 165.5 166 166.5 167 167.5 168 168.5 169 169.5 170 170.5 171 171.5 172 172.5 173 173.5 174 174.5 175 175.5 176 176.5 177 177.5 178 178.5 179 179.5 180 180.5 181 181.5 182 182.5 183 183.5 184 184.5 185 185.5 186 186.5 187 187.5 188 188.5 189 189.5 190 190.5 191 191.5 192 192.5 193 193.5 194 194.5 195 195.5 196 196.5 197 197.5 198 198.5 199 199.5 200 200.5 201 201.5 202 202.5 203 203.5 204 204.5 205 205.5 206 206.5 207 207.5 208 208.5 209 209.5 210 210.5 211 211.5 212 212.5 213 213.5 214 214.5 215 215.5 216 216.5 217 217.5 218 218.5 219 219.5 220 220.5 221 221.5 222 222.5 223 223.5 224 224.5 225 225.5 226 226.5 227 227.5 228 228.5 229 229.5 230 230.5 231 231.5 232 232.5 233 233.5 234 234.5 235 235.5 236 236.5 237 237.5 238 238.5 239 239.5 240 240.5 241 241.5 242 242.5 243 243.5 244 244.5 245 245.5 246 246.5 247 247.5 248 248.5 249 249.5 250 250.5 251 251.5 252 252.5 253 253.5 254 254.5 255 255.5 256 256.5 257 257.5 258 258.5 259 259.5 260 260.5 261 261.5 262 262.5 263 263.5 264 264.5 265 265.5 266 266.5 267 267.5 268 268.5 269 269.5 270 270.5 271 271.5 272 272.5 273 273.5 274 274.5 275 275.5 276 276.5 277 277.5 278 278.5 279 279.5 280 280.5 281 281.5 282 282.5 283 283.5 284 284.5 285 285.5 286 286.5 287 287.5 288 288.5 289 289.5 290 290.5 291 291.5 292 292.5 293 293.5 294 294.5 295 295.5 296 296.5 297 297.5 298 298.5 299 299.5 300 300.5 301 301.5 302 302.5 303 303.5 304 304.5 305 305.5 306 306.5 307 307.5 308 308.5 309 309.5 310 310.5 311 311.5 312 312.5 313 313.5 314 314.5 315 315.5 316 316.5 317 317.5 318 318.5 319 319.5 320 320.5 321 321.5 322 322.5 323 323.5 324 324.5 325 325.5 326 326.5 327 327.5 328 328.5 329 329.5 330 330.5 331 331.5 332 332.5 333 333.5 334 334.5 335 335.5 336 336.5 337 337.5 338 338.5 339 339.5 340 340.5 341 341.5 342 342.5 343 343.5 344 344.5 345 345.5 346 346.5 347 347.5 348 348.5 349 349.5 350 350.5 351 351.5 352 352.5 353 353.5 354 354.5 355 355.5 356 356.5 357 357.5 358 358.5 359 359.5 360 360.5 361 361.5 362 362.5 363 363.5 364 364.5 365 365.5 366 366.5 367 367.5 368 368.5 369 369.5 370 370.5 371 371.5 372 372.5 373 373.5 374 374.5 375 375.5 376 376.5 377 377.5 378 378.5 379 379.5 380 380.5 381 381.5 382 382.5 383 383.5 384 384.5 385 385.5 386 386.5 387 387.5 388 388.5 389 389.5 390 390.5 391 391.5 392 392.5 393 393.5 394 394.5 395 395.5 396 396.5 397 397.5 398 398.5 399 399.5 400 400.5 401 401.5 402 402.5 403 403.5 404 404.5 405 405.5 406 406.5 407 407.5 408 408.5 409 409.5 410 410.5 411 411.5 412 412.5 413 413.5 414 414.5 415 415.5 416 416.5 417 417.5 418 418.5 419 419.5 420 420.5 421 421.5 422 422.5 423 423.5 424 424.5 425 425.5 426 426.5 427 427.5 428 428.5 429 429.5 430 430.5 431 431.5 432 432.5 433 433.5 434 434.5

—

● ● ● ● ● ●

1

2.//16X, 5MS IGM A, 25X, 7MJ-OMEGA, //1)

— — —

● ● ●

1997 1998 1999

• • • •

— — —

[illegible]

• •

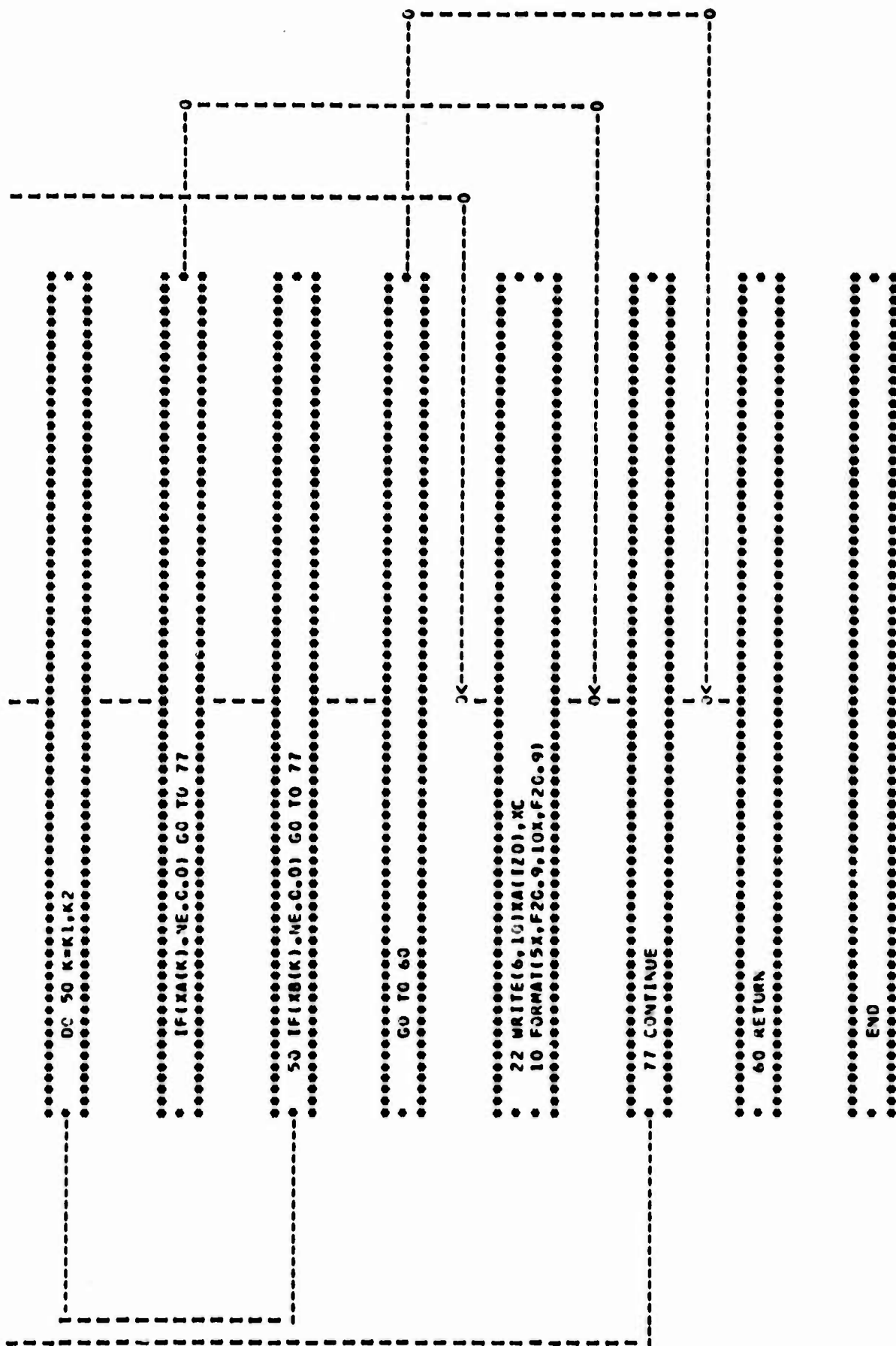
- 1 -

• • •

— — —

• • • • •

— — —



(ENTRANCE)

```

.....
SUBROUTINE FRE.RS(IGMAX,OMEGA)
.....
DIMENSION SAVE1(00,100),SAVE2(100,100),ACOEFF(100),WCOEFF(100),
.....
      AAAAA(9),BRRAB(9),SIGMAX(200),OMEGA(200),DEC(9),
.....
      WEGAS(50),OMEGA(17),W_OG(3000),PWLIN(3000),MDLIN(3000),
.....
      PWPLN(3000),MPDBLN(3000),ALIM(7),BIGSEM(3000),
.....
      EWSUBS(3000)
.....
REAL MDLIN,MPDBLN
.....
INTEGER FWRD,IER(4)
.....
INTEGER PCIN(130,100)
.....
COMMON/FREK/ACOEFF,WCOEFF
.....
COMMON/INER2/ALIM
.....
COMMON/INER3/OMEGA
.....
COMMON/NERA/EWSUBS
.....
COMMON /INF0/ ICOUNT
.....
COMMON SAVE1,SAVE2,PCINT
.....
EQUIVALENCE WCO(1),SAVE1(1),(PWLIN(1),SAVE2(1)),
.....
      1 (SAVE1(1001),MDLIN(1)),(SAVE2(1001),PWPLN(1)),
.....
      2 (SAVE1(1001),MPDBLN(1)),(SAVE2(1001),BIGSEM(1)),
.....
      3 (ALIM(1),XSVAL),(AAAAA(1),ACOEFF(2)),(BRRAB(1),
.....
      4 WCOEFF(2))
.....
TENS(X)=X*10.0
.....
DATA(DEC(1),1,1,7)/1.0,2.0,3.0,4.0,5.0,6.0,7.0,8.0,9.0/
.....
DATA (EY1/4M UB,LE12/4M DEG/
.....
      K100
.....
      KOUNT=1
.....

```

205

```

.....
      NO 61 I=1,4
.....
.....
.....
      41 IEXP(1)=0
.....
.....
.....
      NO 40 I=1,3000
.....
.....
.....
.....
      A2 W_OG(1)=0.0
.....
      PWLIN(1)=0.0
.....
      BIGSEM(1)=0.0
.....
      EWSUBS(1)=0.0
.....
      W_OG(1)=0.0
.....
      PWLIN(1)=0.0
.....

```



```

.....
AN V03MLN(1)Z0.U
.....

```

```

.....
WRITE(6,51)
  REAN(5,10)XSALL,NEGEE,U9LIM
  REAN(5,12)PRUEXP
  IF(PRUEXP.AE.0)REAU(5,12)(IEXP(K),K=1,4)
  12 FORMAT(411H)
  19 FORMAT(3F10.0,50A)
.....

```

```

.....
DO 90 I=2,7
.....

```

```

.....
  30 A I(M(1)=TEASALIN(1-1))
.....

```

```

.....
DO 70 J=1,4
.....

```

```

.....
DO 71 I=1,9
.....

```

```

.....
  K=K+1
.....

```

```

.....
  74 OVERGAS(K)=PEL(1)*ALIM(J)
.....

```

```

.....
70 K1X1=0
.....

.....
OUEGAS(55)=A-LIN(1)
K=1
CALL FLOORB(25,OUEGAS,KOUNT)
OK
.....

.....
41 YEARS(OUEGAS(K))
YEARS(SIGMAX(K))
.....

.....
IF(Y=0.0.0) GO TO 15
.....

.....
VONEBAY
.....

.....
GO TO 20
.....

OK
.....

.....
IF(Y=0.0.0) GO TO 10
.....

.....
VONEBAY
.....

```

```

.....
* 20 CALL SHEPARD(OMEGA)
*   CALL FIGURE(17,OMEGA,KOUNT)
.....

.....
*   IF(KOUNT,GE,2984)GO TO 40
.....

.....
*   GO TO 50
.....

.....
*   IN VARIOUS
*   KOUNT1
.....

.....
*   IF(K1,GT,200) GO TO 40
.....

.....
*   NO 11 LAKKAT
.....

.....
*   YEARS(OMEGA(L))
*   YEARS(SIGMA(L))
.....

.....
*   IF(X,NE,0.0,OR,Y,NE,0.0) GO TO 40
.....

```



```

.....
* 40 WRITE(6,14)
*   CALL NEWBY(NEUR2E,MLUG,MLIN,LE12)
.....

```

```

.....
*   IF(LEVP(1),50,0) GO TO 30
.....

```

```

.....
*   JSTEP(1)
.....

```

```

.....
*   DO 14 I=1,JJ
.....

```

```

.....
*   14 CALL EXPBY(MLUG,3000,MLIN,3000,DEGSE,LE12)
.....

```

```

.....
*   10 CALL NEWBY(MLIN,MLUG,MLIN,LE11)
.....

```

```

.....
*   IF(LEVP(2),50,0) GO TO 31
.....

```

```

.....
*   JSTEP(2)
.....

```

```

.....
*   DO 14 I=1,JJ
.....

```

* W 74L 240 CYCLO. 3000, 4000 L.A. 3000, 4000 L.F.V)

[illegible][illegible]

100

• NO VA 19.000 •

(L87-3303C; U00TIN) 19-00N-DIJAURE IV VA

AS 221 NEW YORK 14, N.Y. 10014, N.Y. 10014

• 12114424,64.00 10 33

(P) A. C.

•

•

[illegible]

100

100

[illegible]

SECRET

213

21 CONTINUE

RETURN

END

[illegible]

END

(ENTRANCE)

```
.....  
C .....  
C .....  
C SUBROUTINE LOGNET(AX,LIMITM)  
C DIMENSION ALIM(1),DIMS(20)  
C EQUIVALENCE(WASTE,AXX)  
C COMMON/INF02/ALIM  
C TENS(X)X*10.0  
C DATA (DIMS(1),1,1,1,20) /1.0,1.15,1.3,1.5,1.7,1.9,2.25,2.5,2.85,  
C 3.15,3.5,3.95,4.5,5.05,5.4,6.05,6.8,7.7,8.8,9.99999/  
C LIMITM=0  
C LIMITX=0  
C LIMTY=0  
.....
```

I
I
I

```
.....  
C NO 20 1,1,1,1,1  
.....
```

I
I
I

```
.....  
C AAB2(1-1)  
.....
```

I
I
I

```
.....  
C 20 IF(WX.LT.ALM(1-1).AND.WX.GE.ALIM(1)) GO TO 21  
.....
```

I
I
I

```
.....  
C RETURN  
.....
```

```
.....  
C 21 LIMITYSTENS(AA,  
C WASTEARS(WX)  
C IF(WASTE.EQ.U.U) RETURN  
.....
```

I
0
I

```
.....  
C 20 IF(AXX.GE.1.U.ANU.AXX.LT.10.0) GO TO 22  
.....
```

I
I
I

• END •

(ENTRANCE)

```
.....  
C  
C  
SUBROUTINE BLANK (POINT,N1,N2,LIM1)  
INTEGER POINT(N1,N2),BLANK,DASH,3  
COMMON /INFO/ O,BLANK,DASH  
DATA BLANK/1/,DASH/10-/,O/10/
```

```
.....  
DO 51 M=1,A1
```

```
.....  
DO 51 N=1,A2
```

```
.....  
51 POINT(M,N)=BLANK
```

221

```
.....  
DO 52 M=1,A1
```

```
.....  
POINT(M,1)=DASH
```

```
.....  
52 POINT(M,N)=DASH
```

```
.....  
DO 53 M=1,A2
```



```

.....
      POINT(1,M)G
.....
.....
      XX POINT(LIM,M)G
.....
.....
      RETURN
.....
.....
      END
.....

```

(ENTRANCE)

```
.....
C.....
C.....
SUBROUTINE NEWPL(A,Y,Z,LET)
  INTEGER B(1)(130,100),B(1)NF(120,50),STAB,0
  DIMENSION ALIM(7)
  DIMENSION V(300),Z(300)
  DIMENSION SAV1(100,100),SAVE2(100,100)
  COMMON /INFO/ O,BLANK,DASH
  COMMON /INFO2/ ALIN
  COMMON SAVE1,SAVE2,B(1)NF
  EQUIVALENCE(B(1),1),B(1)NF(1,1)
  DATA STAB/10/,O/1M/
  * FORMAT(1M)
  CALL BLANKR(POINIF,120,50,120)
.....
.....
      GO TO 11,3000
.....
.....
      IF(V(1),EQ,0.0) GO TO 11
.....
.....
      GO TO 12
.....
.....
      IF(V(1),EQ,0.0) GO TO 10
.....
.....
      IF(V(1),EQ,0.0) GO TO 10
.....
.....
      IF(V(1),EQ,0.0) GO TO 10
.....
      CALL LOOFR(1),LIMIT
.....
.....
      IF(V(1),EQ,0.0) GO TO 10
.....
.....
```

```

.....
CALL IAEAR(A,IMTV,2(1))
.....

```

```

.....
IF(IMTV.EQ.0) GO TO 10
.....

```

```

.....
POINTF(IMTV,IMTV)ESTAR
.....

```

```

.....
GO
.....

```

```

.....
DO 40 I=20,100,20
.....

```

```

.....
DO 40 J=1,50
.....

```

```

.....
40 POINTF(I,J)EU
.....

```

```

.....
WRITE(6,1)
DO 50 FORMAT(1X,A10F50.1,1X,3MM/5.0X),10F50.1)
DO 50
WRITE(6,51)JULET,(POINTF(I,1),I=1,120)
51 FORMAT(1X,10, A4.1A,120A1)
WRITE(6,52)(POINTF(I,J),I=1,120),J=2,57)
DO 50
WRITE(6,51)JULET,(POINTF(I,50),I=1,120)
50 FORMAT(10X,100X)
WRITE(6,50)ALLM
.....

```

REVIEW

END

```

SUBROUTINE LINEAR(A,L,M1,V,V)
DIMENSION ENOBT(10)
COMMON
ENOBTS(10)
ENOBTS(10)

```

1
 2
 3
 4
 5
 6
 7
 8
 9
 10
 11
 12
 13
 14
 15
 16
 17
 18
 19
 20
 21
 22
 23
 24
 25
 26
 27
 28
 29
 30
 31
 32
 33
 34
 35
 36
 37
 38
 39
 40
 41
 42
 43
 44
 45
 46
 47
 48
 49
 50
 51
 52
 53
 54
 55
 56
 57
 58
 59
 60
 61
 62
 63
 64
 65
 66
 67
 68
 69
 70
 71
 72
 73
 74
 75
 76
 77
 78
 79
 80
 81
 82
 83
 84
 85
 86
 87
 88
 89
 90
 91
 92
 93
 94
 95
 96
 97
 98
 99
 100
 101
 102
 103
 104
 105
 106
 107
 108
 109
 110
 111
 112
 113
 114
 115
 116
 117
 118
 119
 120
 121
 122
 123
 124
 125
 126
 127
 128
 129
 130
 131
 132
 133
 134
 135
 136
 137
 138
 139
 140
 141
 142
 143
 144
 145
 146
 147
 148
 149
 150
 151
 152
 153
 154
 155
 156
 157
 158
 159
 160
 161
 162
 163
 164
 165
 166
 167
 168
 169
 170
 171
 172
 173
 174
 175
 176
 177
 178
 179
 180
 181
 182
 183
 184
 185
 186
 187
 188
 189
 190
 191
 192
 193
 194
 195
 196
 197
 198
 199
 200
 201
 202
 203
 204
 205
 206
 207
 208
 209
 210
 211
 212
 213
 214
 215
 216
 217
 218
 219
 220
 221
 222
 223
 224
 225
 226
 227
 228
 229
 230
 231
 232
 233
 234
 235
 236
 237
 238
 239
 240
 241
 242
 243
 244
 245
 246
 247
 248
 249
 250
 251
 252
 253
 254
 255
 256
 257
 258
 259
 260
 261
 262
 263
 264
 265
 266
 267
 268
 269
 270
 271
 272
 273
 274
 275
 276
 277
 278
 279
 280
 281
 282
 283
 284
 285
 286
 287
 288
 289
 290
 291
 292
 293
 294
 295
 296
 297
 298
 299
 300
 301
 302
 303
 304
 305
 306
 307
 308
 309
 310
 311
 312
 313
 314
 315
 316
 317
 318
 319
 320
 321
 322
 323
 324
 325
 326
 327
 328
 329
 330
 331
 332
 333
 334
 335
 336
 337
 338
 339
 340
 341
 342
 343
 344
 345
 346
 347
 348
 349
 350
 351
 352
 353
 354
 355
 356
 357
 358
 359
 360
 361
 362
 363
 364
 365
 366
 367
 368
 369
 370
 371
 372
 373
 374
 375
 376
 377
 378
 379
 380
 381
 382
 383
 384
 385
 386
 387
 388
 389
 390
 391
 392
 393
 394
 395
 396
 397
 398
 399
 400
 401
 402
 403
 404
 405
 406
 407
 408
 409
 410
 411
 412
 413
 414
 415
 416
 417
 418
 419
 420
 421
 422
 423
 424
 425
 426
 427
 428
 429
 430
 431
 432
 433
 434
 435
 436
 437
 438
 439
 440
 441
 442
 443
 444
 445
 446
 447
 448
 449
 450
 451
 452
 453
 454
 455
 456
 457
 458
 459
 460
 461
 462
 463
 464
 465
 466
 467
 468
 469
 470
 471
 472
 473
 474
 475
 476
 477
 478
 479
 480
 481
 482
 483
 484
 485
 486
 487
 488
 489
 490
 491
 492
 493
 494
 495
 496
 497
 498
 499
 500
 501
 502
 503
 504
 505
 506
 507
 508
 509
 510
 511
 512
 513
 514
 515
 516
 517
 518
 519
 520
 521
 522
 523
 524
 525

'C ENEMY () ENEMY (-) DE LYA

ENCLAVE (199) 00A

.....

 * FIVE EIGHTY TWO AND A HALF PERCENT (11.100) PA 2A

PAGE

14-00000

1 1 0
1 1 0
1 1 0

.....
4n RETURN
.....

.....
END
.....

```

.....
C .....
C .....
SUBROUTINE SWEEP(XOMEGA)
DIMENSION PVEGA(17),SWEPER(17)
COMMON/INF03/Omega
DATA (SWEPER(I),I=1,17) /0.01,0.05,0.1,0.2,0.3,0.4,0.5,0.6,0.7,
    .8,0.9,1.0,2.0,3.0,6.0,10.0/
      I
      I
DO 70 K=1,17
      I
      I
      I
      I
      I
70 OMEGA(K)= SWEPER(K)*XOMEGA
      I
      I
      I
RETURN
.....
END
.....

```

<p>AD _____</p> <p>A LINEAR AND NONLINEAR SYSTEMS ANALYSIS TOOL: LINE PRINTER PLOTS OF CHARACTERISTIC EQUATION ROOT LOCI, BODE AND POPOV PLOTS OF SYSTEM TRANSFER FUNCTIONS</p> <p>USA Aberdeen Research and Development Center, Army Materiel Systems Analysis Agency, Aberdeen Proving Ground, Md. 21005</p> <p>Army Materiel Systems Analysis Agency Technical Memorandum No. 69, UNCLASSIFIED Report</p> <p>A LINEAR AND NONLINEAR SYSTEMS ANALYSIS TOOL: LINE PRINTER PLOTS OF CHARACTERISTIC EQUATION ROOT LOCI, BODE AND POPOV PLOTS OF SYSTEM TRANSFER FUNCTIONS, Harold H. Burke, Robert L. Payne, Jr., AMSAA. DDC; Project No. RDT&E 1P765801M1102, March 1970</p> <p><u>Trace Index Terms</u></p> <p>Systems analyses, computer capability Phase-amplitude characteristic, nonlinear systems Linear frequency response method Closed loop transfer function Nonlinear frequency response method Bode plots, system transfer Open loop transfer function Popov plots, system transfer Line printer plots, root loci</p> <p>1. Systems analyses, computer capability 2. Linear frequency response method 3. Nonlinear frequency response method 4. Line printer plots, root loci</p> <p>A LINEAR AND NONLINEAR SYSTEMS ANALYSIS TOOL: LINE PRINTER PLOTS OF CHARACTERISTIC EQUATION ROOT LOCI, BODE AND POPOV PLOTS OF SYSTEM TRANSFER FUNCTIONS, Harold H. Burke, Robert L. Payne, Jr.</p>	<p>Burke, Harold H., Payne, Robert L., Jr.</p> <p>AD _____</p> <p>USA Aberdeen Research and Development Center, Army Materiel Systems Analysis Agency, Aberdeen Proving Ground, Md. 21005</p> <p>Army Materiel Systems Analysis Agency Technical Memorandum No. 69, UNCLASSIFIED Report</p> <p>A LINEAR AND NONLINEAR SYSTEMS ANALYSIS TOOL: LINE PRINTER PLOTS OF CHARACTERISTIC EQUATION ROOT LOCI, BODE AND POPOV PLOTS OF SYSTEM TRANSFER FUNCTIONS, Harold H. Burke, Robert L. Payne, Jr., AMSAA. DDC; Project No. RDT&E 1P765801M1102, March 1970</p> <p><u>Trace Index Terms</u></p> <p>Systems analyses, computer capability Phase-amplitude characteristic, nonlinear systems Linear frequency response method Closed loop transfer function Nonlinear frequency response method Bode plots, system transfer Open loop transfer function Popov plots, system transfer Line printer plots, root loci</p> <p>1. Systems analyses, computer capability 2. Linear frequency response method 3. Nonlinear frequency response method 4. Line printer plots, root loci</p> <p>A LINEAR AND NONLINEAR SYSTEMS ANALYSIS TOOL: LINE PRINTER PLOTS OF CHARACTERISTIC EQUATION ROOT LOCI, BODE AND POPOV PLOTS OF SYSTEM TRANSFER FUNCTIONS, Harold H. Burke, Robert L. Payne, Jr.</p>
<p>AD _____</p> <p>A LINEAR AND NONLINEAR SYSTEMS ANALYSIS TOOL: LINE PRINTER PLOTS OF CHARACTERISTIC EQUATION ROOT LOCI, BODE AND POPOV PLOTS OF SYSTEM TRANSFER FUNCTIONS, Harold H. Burke, Robert L. Payne, Jr., AMSAA. DDC; Project No. RDT&E 1P765801M1102, March 1970</p> <p><u>Trace Index Terms</u></p> <p>Systems analyses, computer capability Phase-amplitude characteristic, nonlinear systems Linear frequency response method Closed loop transfer function Nonlinear frequency response method Bode plots, system transfer Open loop transfer function Popov plots, system transfer Line printer plots, root loci</p> <p>1. Systems analyses, computer capability 2. Linear frequency response method 3. Nonlinear frequency response method 4. Line printer plots, root loci</p> <p>A LINEAR AND NONLINEAR SYSTEMS ANALYSIS TOOL: LINE PRINTER PLOTS OF CHARACTERISTIC EQUATION ROOT LOCI, BODE AND POPOV PLOTS OF SYSTEM TRANSFER FUNCTIONS, Harold H. Burke, Robert L. Payne, Jr.</p>	<p>AD _____</p> <p>Linear frequency response method</p> <p>USA Aberdeen Research and Development Center, Army Materiel Systems Analysis Agency, Aberdeen Proving Ground, Md. 21005</p> <p>Army Materiel Systems Analysis Agency Technical Memorandum No. 69, UNCLASSIFIED Report</p> <p>A LINEAR AND NONLINEAR SYSTEMS ANALYSIS TOOL: LINE PRINTER PLOTS OF CHARACTERISTIC EQUATION ROOT LOCI, BODE AND POPOV PLOTS OF SYSTEM TRANSFER FUNCTIONS, Harold H. Burke, Robert L. Payne, Jr., AMSAA. DDC; Project No. RDT&E 1P765801M1102, March 1970</p> <p><u>Trace Index Terms</u></p> <p>Systems analyses, computer capability Phase-amplitude characteristic, nonlinear systems Linear frequency response method Closed loop transfer function Nonlinear frequency response method Bode plots, system transfer Open loop transfer function Popov plots, system transfer Line printer plots, root loci</p> <p>1. Systems analyses, computer capability 2. Linear frequency response method 3. Nonlinear frequency response method 4. Line printer plots, root loci</p> <p>A LINEAR AND NONLINEAR SYSTEMS ANALYSIS TOOL: LINE PRINTER PLOTS OF CHARACTERISTIC EQUATION ROOT LOCI, BODE AND POPOV PLOTS OF SYSTEM TRANSFER FUNCTIONS, Harold H. Burke, Robert L. Payne, Jr.</p>

AD _____ A LINEAR AND NONLINEAR SYSTEMS ANALYSIS TOOL: LINE PRINTER
PLOTS OF CHARACTERISTIC EQUATION ROOT LOCI, BODE AND POPOV
PLOTS OF SYSTEM TRANSFER FUNCTIONS
USA Aberdeen Research and Development Center, Army Materiel Systems Analysis Agency,
Aberdeen Proving Ground, Md. 21005

Army Materiel Systems Analysis Agency Technical Memorandum No. 69, UNCLASSIFIED Report

A LINEAR AND NONLINEAR SYSTEMS ANALYSIS TOOL: LINE PRINTER PLOTS OF CHARACTERISTIC EQUATION ROOT LOCI, BODE AND POPOV PLOTS OF SYSTEM TRANSFER FUNCTIONS, Harold H. Burke, Robert L. Payne, Jr., AMEAA. DDC: Project No. RDT&E 1P765801M1102.

March 1970

Trace Index Terms

Systems analyses, computer capability
Linear frequency response method
Nonlinear frequency response method
Open loop transfer function
Line printer plots, root loci
Phase-amplitude characteristic, nonlinear systems
Closed loop transfer function
Bode plots, system transfer
Popov plots, system transfer

1. Systems analyses, computer capability
 2. Linear frequency response method
 3. Nonlinear frequency response method
 4. Line printer plots, root loci
- EQUATION ROOT LOCI, BODE AND POPOV PLOTS OF SYSTEM TRANSFER FUNCTIONS, Harold H. Burke, Robert L. Payne, Jr.

AD _____ Systems analyses, computer capability

USA Aberdeen Research and Development Center, Army Materiel Systems Analysis Agency,
Aberdeen Proving Ground, Md. 21005

Army Materiel Systems Analysis Agency Technical Memorandum No. 69, UNCLASSIFIED Report
A LINEAR AND NONLINEAR SYSTEMS ANALYSIS TOOL: LINE PRINTER PLOTS OF CHARACTERISTIC EQUATION ROOT LOCI, BODE AND POPOV PLOTS OF SYSTEM TRANSFER FUNCTIONS, Harold H. Burke, Robert L. Payne, Jr., AM6AA. DDC; Project No. RDT&E 1P765801M1102,
March 1970

Trace Index Terms

Systems analyses, computer capability
Linear frequency response method
Nonlinear frequency response method
Open loop transfer function
Line printer plots, root loci
Phase-amplitude characteristic, nonlinear systems
Closed loop transfer function
Bode plots, system transfer
Polar plots, system transfer

1. Systems analyses, computer capability
 2. Linear frequency response method
 3. Nonlinear frequency response method
 4. Line printer plots, root loci
- A LINEAR AND NONLINEAR SYSTEMS ANALYSIS TOOL: LINE PRINTER PLOTS OF CHARACTERISTIC EQUATION ROOT LOCI, BODE AND POPOV PLOTS OF SYSTEM TRANSFER FUNCTIONS, HAROLD H. BURKE, ROBERT L. PAYNE, JR.

AD _____
Burke, Harold H., Payne, Robert L., Jr.
USA Aberdeen Research and Development Center, Army Materiel Systems Analysis Agency,
Aberdeen Proving Ground, Md. 21005

Army Materiel Systems Analysis Agency Technical Memorandum No. 69, UNCLASSIFIED Report

A LINEAR AND NONLINEAR SYSTEMS ANALYSIS TOOL: LINE PRINTER PLOTS OF CHARACTERISTIC EQUATION ROOT LOCI, BUDE AND POPOV PLOTS OF SYSTEM TRANSFER FUNCTIONS, HAROLD H. Burke, Robert L. Payne, Jr., ANSAA. DDC: Project No. RDT&E 1P765801M102.

March 1970

Trace Index Terms

Systems analyses, computer capability
Linear frequency response method
Nonlinear frequency response method
Open loop transfer function
Line printer plots, root loci
Phase-amplitude characteristic, nonlinear systems
Closed loop transfer function
Bode plots, system transfer
Popov plots, system transfer

1. Systems analyses, computer capability
 2. Linear frequency response method
 3. Nonlinear frequency response method
 4. Line printer plots, root loci
- A LINEAR AND NONLINEAR SYSTEMS ANALYSIS TOOL: LINE PRINTER PLOTS OF CHARACTERISTIC EQUATION ROOT LOCI, BODE AND POPOV PLOTS OF SYSTEM TRANSFER FUNCTIONS Harold H. Burke, Robert L. Payne, Jr.

AD _____

Army Materiel Systems Analysis Agency Technical Memorandum No. 69, UNCLASSIFIED
 Report
 A LINEAR AND NONLINEAR SYSTEMS ANALYSIS TOOL: LINE PRINTER PLOTS OF CHARACTERISTIC
 EQUATION ROOT LOCI, BODE AND POPOV PLOTS OF SYSTEM TRANSFER FUNCTIONS, Harold H.
 Burke, Robert L. Payne, Jr., AMEAA. DDC; Project No. RDT&E 1F765801M1102,
 March 1970

Trace Index Terms

Systems analyses, computer capability
Linear frequency response method
Nonlinear frequency response method
Open loop transfer function
Line printer plots, root loci
Phase-amplitude characteristic,
nonlinear systems
Closed loop transfer function
Bode plots, system transfer
Pencil plots, system transfer

1. Systems analyses, computer capability
 2. Linear frequency response method
 3. Nonlinear frequency response method
 4. Line printer plots, root loci
- A LINEAR AND NONLINEAR SYSTEMS ANALYSIS TOOL: LINE PRINTER PLOTS OF CHARACTERISTIC EQUATION ROOT LOCI, BODE AND POPOV PLOTS OF SYSTEM TRANSFER FUNCTIONS Harold H. Burke, Robert L. Payne, Jr.

AD	Nonlinear frequency response method	Line printer plots, root loci
<p>USA Aberdeen Research and Development Center, Army Materiel Systems Analysis Agency, Aberdeen Proving Ground, Md. 21005</p> <p>Army Materiel Systems Analysis Agency Technical Memorandum No. 69, UNCLASSIFIED Report</p> <p>A LINEAR AND NONLINEAR SYSTEMS ANALYSIS TOOL: LINE PRINTER PLOTS OF CHARACTERISTIC EQUATION ROOT LOCI, BODE AND POPOV PLOTS OF SYSTEM TRANSFER FUNCTIONS, Harold H. Burke, Robert L. Payne, Jr., AMSAA. DDC; Project No. RDT&E 1P765801M1102, March 1970</p> <p><u>Trace Index Terms</u></p> <p>Systems analyses, computer capability Phase-amplitude characteristic, nonlinear systems</p> <p>Linear frequency response method Closed loop transfer function</p> <p>Nonlinear frequency response method Bode plots, system transfer</p> <p>Open loop transfer function Popov plots, system transfer</p> <p>Line printer plots, root loci</p> <p>1. Systems analyses, computer capability 2. Linear frequency response method</p> <p>3. Nonlinear frequency response method 4. Line printer plots, root loci</p> <p>A LINEAR AND NONLINEAR SYSTEMS ANALYSIS TOOL: LINE PRINTER PLOTS OF CHARACTERISTIC EQUATION ROOT LOCI, BODE AND POPOV PLOTS OF SYSTEM TRANSFER FUNCTIONS, Harold H. Burke, Robert L. Payne, Jr.</p>		
AD	Nonlinear frequency response method	Line printer plots, root loci
<p>USA Aberdeen Research and Development Center, Army Materiel Systems Analysis Agency, Aberdeen Proving Ground, Md. 21005</p> <p>Army Materiel Systems Analysis Agency Technical Memorandum No. 69, UNCLASSIFIED Report</p> <p>A LINEAR AND NONLINEAR SYSTEMS ANALYSIS TOOL: LINE PRINTER PLOTS OF CHARACTERISTIC EQUATION ROOT LOCI, BODE AND POPOV PLOTS OF SYSTEM TRANSFER FUNCTIONS, Harold H. Burke, Robert L. Payne, Jr., AMSAA. DDC; Project No. RDT&E 1P765801M1102, March 1970</p> <p><u>Trace Index Terms</u></p> <p>Systems analyses, computer capability Phase-amplitude characteristic, nonlinear systems</p> <p>Linear frequency response method Closed loop transfer function</p> <p>Nonlinear frequency response method Bode plots, system transfer</p> <p>Open loop transfer function Popov plots, system transfer</p> <p>Line printer plots, root loci</p> <p>1. Systems analyses, computer capability 2. Linear frequency response method</p> <p>3. Nonlinear frequency response method 4. Line printer plots, root loci</p> <p>A LINEAR AND NONLINEAR SYSTEMS ANALYSIS TOOL: LINE PRINTER PLOTS OF CHARACTERISTIC EQUATION ROOT LOCI, BODE AND POPOV PLOTS OF SYSTEM TRANSFER FUNCTIONS, Harold H. Burke, Robert L. Payne, Jr.</p>		
AD	Nonlinear frequency response method	Line printer plots, root loci
<p>USA Aberdeen Research and Development Center, Army Materiel Systems Analysis Agency, Aberdeen Proving Ground, Md. 21005</p> <p>Army Materiel Systems Analysis Agency Technical Memorandum No. 69, UNCLASSIFIED Report</p> <p>A LINEAR AND NONLINEAR SYSTEMS ANALYSIS TOOL: LINE PRINTER PLOTS OF CHARACTERISTIC EQUATION ROOT LOCI, BODE AND POPOV PLOTS OF SYSTEM TRANSFER FUNCTIONS, Harold H. Burke, Robert L. Payne, Jr., AMSAA. DDC; Project No. RDT&E 1P765801M1102, March 1970</p> <p><u>Trace Index Terms</u></p> <p>Systems analyses, computer capability Phase-amplitude characteristic, nonlinear systems</p> <p>Linear frequency response method Closed loop transfer function</p> <p>Nonlinear frequency response method Bode plots, system transfer</p> <p>Open loop transfer function Popov plots, system transfer</p> <p>Line printer plots, root loci</p> <p>1. Systems analyses, computer capability 2. Linear frequency response method</p> <p>3. Nonlinear frequency response method 4. Line printer plots, root loci</p> <p>A LINEAR AND NONLINEAR SYSTEMS ANALYSIS TOOL: LINE PRINTER PLOTS OF CHARACTERISTIC EQUATION ROOT LOCI, BODE AND POPOV PLOTS OF SYSTEM TRANSFER FUNCTIONS, Harold H. Burke, Robert L. Payne, Jr.</p>		

<p>AD _____</p> <p>Nonlinear frequency response method</p> <p>USA Aberdeen Research and Development Center, Army Materiel Systems Analysis Agency, Aberdeen Proving Ground, Md. 21005</p> <p>Army Materiel Systems Analysis Agency Technical Memorandum No. 69, UNCLASSIFIED Report</p> <p>A LINEAR AND NONLINEAR SYSTEMS ANALYSIS TOOL: LINE PRINTER PLOTS OF CHARACTERISTIC EQUATION ROOT LOCI, BODE AND POPOV PLOTS OF SYSTEM TRANSFER FUNCTIONS, Harold H. Burke, Robert L. Payne, Jr., ANSAA. DDC; Project No. RDT&E 1P765801M1102, March 1970</p> <p><u>Trace Index Terms</u></p> <p>Systems analyses, computer capability Phase-amplitude characteristic, nonlinear systems</p> <p>Linear frequency response method Closed loop transfer function</p> <p>Nonlinear frequency response method Bode plots, system transfer</p> <p>Open loop transfer function Popov plots, system transfer</p> <p>Line printer plots, root loci</p> <p>1. Systems analyses, computer capability 2. Linear frequency response method</p> <p>3. Nonlinear frequency response method 4. Line printer plots, root loci</p> <p>A LINEAR AND NONLINEAR SYSTEMS ANALYSIS TOOL: LINE PRINTER PLOTS OF CHARACTERISTIC EQUATION ROOT LOCI, BODE AND POPOV PLOTS OF SYSTEM TRANSFER FUNCTIONS, Harold H. Burke, Robert L. Payne, Jr.</p>	<p>AD _____</p> <p>Line printer plots, root loci</p> <p>USA Aberdeen Research and Development Center, Army Materiel Systems Analysis Agency, Aberdeen Proving Ground, Md. 21005</p> <p>Army Materiel Systems Analysis Agency Technical Memorandum No. 69, UNCLASSIFIED Report</p> <p>A LINEAR AND NONLINEAR SYSTEMS ANALYSIS TOOL: LINE PRINTER PLOTS OF CHARACTERISTIC EQUATION ROOT LOCI, BODE AND POPOV PLOTS OF SYSTEM TRANSFER FUNCTIONS, Harold H. Burke, Robert L. Payne, Jr., ANSAA. DDC; Project No. RDT&E 1P765801M1102, March 1970</p> <p><u>Trace Index Terms</u></p> <p>Systems analyses, computer capability Phase-amplitude characteristic, nonlinear systems</p> <p>Linear frequency response method Closed loop transfer function</p> <p>Nonlinear frequency response method Bode plots, system transfer</p> <p>Open loop transfer function Popov plots, system transfer</p> <p>Line printer plots, root loci</p> <p>1. Systems analyses, computer capability 2. Linear frequency response method</p> <p>3. Nonlinear frequency response method 4. Line printer plots, root loci</p> <p>A LINEAR AND NONLINEAR SYSTEMS ANALYSIS TOOL: LINE PRINTER PLOTS OF CHARACTERISTIC EQUATION ROOT LOCI, BODE AND POPOV PLOTS OF SYSTEM TRANSFER FUNCTIONS, Harold H. Burke, Robert L. Payne, Jr.</p>
<p>AD _____</p> <p>Nonlinear frequency response method</p> <p>USA Aberdeen Research and Development Center, Army Materiel Systems Analysis Agency, Aberdeen Proving Ground, Md. 21005</p> <p>Army Materiel Systems Analysis Agency Technical Memorandum No. 69, UNCLASSIFIED Report</p> <p>A LINEAR AND NONLINEAR SYSTEMS ANALYSIS TOOL: LINE PRINTER PLOTS OF CHARACTERISTIC EQUATION ROOT LOCI, BODE AND POPOV PLOTS OF SYSTEM TRANSFER FUNCTIONS, Harold H. Burke, Robert L. Payne, Jr., ANSAA. DDC; Project No. RDT&E 1P765801M1102, March 1970</p> <p><u>Trace Index Terms</u></p> <p>Systems analyses, computer capability Phase-amplitude characteristic, nonlinear systems</p> <p>Linear frequency response method Closed loop transfer function</p> <p>Nonlinear frequency response method Bode plots, system transfer</p> <p>Open loop transfer function Popov plots, system transfer</p> <p>Line printer plots, root loci</p> <p>1. Systems analyses, computer capability 2. Linear frequency response method</p> <p>3. Nonlinear frequency response method 4. Line printer plots, root loci</p> <p>A LINEAR AND NONLINEAR SYSTEMS ANALYSIS TOOL: LINE PRINTER PLOTS OF CHARACTERISTIC EQUATION ROOT LOCI, BODE AND POPOV PLOTS OF SYSTEM TRANSFER FUNCTIONS, Harold H. Burke, Robert L. Payne, Jr.</p>	<p>AD _____</p> <p>Line printer plots, root loci</p> <p>USA Aberdeen Research and Development Center, Army Materiel Systems Analysis Agency, Aberdeen Proving Ground, Md. 21005</p> <p>Army Materiel Systems Analysis Agency Technical Memorandum No. 69, UNCLASSIFIED Report</p> <p>A LINEAR AND NONLINEAR SYSTEMS ANALYSIS TOOL: LINE PRINTER PLOTS OF CHARACTERISTIC EQUATION ROOT LOCI, BODE AND POPOV PLOTS OF SYSTEM TRANSFER FUNCTIONS, Harold H. Burke, Robert L. Payne, Jr., ANSAA. DDC; Project No. RDT&E 1P765801M1102, March 1970</p> <p><u>Trace Index Terms</u></p> <p>Systems analyses, computer capability Phase-amplitude characteristic, nonlinear systems</p> <p>Linear frequency response method Closed loop transfer function</p> <p>Nonlinear frequency response method Bode plots, system transfer</p> <p>Open loop transfer function Popov plots, system transfer</p> <p>Line printer plots, root loci</p> <p>1. Systems analyses, computer capability 2. Linear frequency response method</p> <p>3. Nonlinear frequency response method 4. Line printer plots, root loci</p> <p>A LINEAR AND NONLINEAR SYSTEMS ANALYSIS TOOL: LINE PRINTER PLOTS OF CHARACTERISTIC EQUATION ROOT LOCI, BODE AND POPOV PLOTS OF SYSTEM TRANSFER FUNCTIONS, Harold H. Burke, Robert L. Payne, Jr.</p>

Unclassified
Security Classification

DOCUMENT CONTROL DATA - R & D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY (Corporate author) USA Aberdeen Research and Development Center, Army Materiel Systems Analysis Agency, Aberdeen Proving Ground, Maryland 21005		2a. REPORT SECURITY CLASSIFICATION Unclassified	
		2b. GROUP	
3. REPORT TITLE A LINEAR AND NONLINEAR SYSTEMS ANALYSIS TOOL: LINE PRINTER PLOTS OF CHARACTERISTIC EQUATION ROOT LOCI, BODE AND POPOV PLOTS OF SYSTEM TRANSFER FUNCTIONS			
4. DESCRIPTIVE NOTES (Type of report and inclusive dates)			
5. AUTHOR(S) (First name, middle initial, last name) Harold H. Burke Robert L. Payne, Jr.			
6. REPORT DATE March 1970		7a. TOTAL NO. OF PAGES 230	7b. NO. OF REFS 9
8a. CONTRACT OR GRANT NO. a. PROJECT NO. RDT&E 1P765801ML102 c. d.		8b. ORIGINATOR'S REPORT NUMBER(S) AMSAA TM-69 8c. OTHER REPORT NO(S) (Any other numbers that may be assigned this report)	
10. DISTRIBUTION STATEMENT This document may be further distributed by the holder only with specific prior approval of the Commanding General, U.S. Army Materiel Command, ATTN: AMCPA-SA, Washington, D.C. 20315.			
11. SUPPLEMENTARY NOTES		12. SPONSORING MILITARY ACTIVITY U.S. Army Materiel Command Washington, D.C.	
13. ABSTRACT A computer program to perform dynamic systems analyses and plot the results is presented. Linear systems and a large classification of nonlinear systems representing engineering, scientific, and economic disciplines can be modeled to permit application of the computer program. Two examples are given to demonstrate the capabilities of the analysis tool. The mathematical model of a missile guidance and control system is analyzed and a ratio of polynomials representing the closed loop transfer function of a high performance model follower aircraft is evaluated. Linear differential equations to the 100th order having real or complex roots can be studied. System characteristic equation root loci and system transfer functions are plotted.			

DD FORM 1473 REPLACES DD FORM 1473, 1 JAN 64, WHICH IS OBSOLETE FOR ARMY USE.

Unclassified
Security Classification

14.	KEY WORDS	LINK A		LINK B		LINK C	
		ROLE	WT	ROLE	WT	ROLE	WT
	Systems analysis						
	Systems synthesis						
	Root locus						
	Bode plots						
	Steady state response						
	Popov plots						
	Modified phase-amplitude characteristic						